High Performance Computing Wales

HPC User Guide

Version 2.2

March 2013

Ewrop & Chymru:
Buddsoddi yn eich dyfodol
Cronfa Datblygu Rhanbarthol Ewrop
Europe & Wales:
Investing in your future
European Regional Development Fund
ERDF
Llywodraeth Cymru
Welsh Government

Table of Contents

1 An Introduction to the User Guide 11

2 An Overview of the HPC Wales System 12

2.1 Collaborative working and User Productivity 13

2.2 The HPC Wales System Architecture 13

2.2.1 High Level Design 14

2.3 System Software Environment and Usage Model 15

2.4 HPC Wales Computing and Networking Infrastructure 16

3 Using the HPC Wales Systems – First Steps 17

3.1 Support 17

3.2 Requesting an account 17

3.3 Accessing the HPC Wales Systems 18

3.3.1 Gaining Access from a UNIX environment 18

3.3.2 Gaining Access from a Windows environment 20

3.3.3 Password Specification 20

3.4 File Transfer 20

3.4.1 Transferring Files from a UNIX environment 20

3.4.2 Transferring Files from a WINDOWS environment 21

4 The Cardiff Hub & Tier-1 Infrastructures 23

4.1 The Cardiff Infrastructure 23

4.1.1 The Cardiff HTC Cluster 24

4.2 Filesystems 25

4.3 The Tier-1 Infrastructure at Aberystwyth, Bangor, and Glamorgan 26

4.4 An Introduction to Using the Linux Clusters 27

4.5 Accessing the Clusters 28

4.5.1 Logging In 28

4.5.2 File Transfer 28

5 The User Environment 30

5.1 Unix shell 30

5.2 Environment variables 30

5.3 Startup scripts 30

5.4 Environment Modules 32

5.4.1 List Available Modules 32

5.4.2 Show Module Information 34
5.4.3 Loading Modules 34
5.4.4 Unloading Modules 34
5.4.5 Verify Currently Loaded Modules 35
5.4.6 Compatibility of Modules 35
5.4.7 Other Module Commands 35
6 Compiling Code on HPC Wales 37
6.1 Details of the available compilers and how to use them 37
6.2 GNU Compiler Collection 37
6.2.1 Documentation 37
6.3 Intel Compilers 37
6.4 Compiling Code – a Simple Example 38
6.5 Libraries 39
6.6 Performance Libraries 40
6.6.1 Math Kernel Library (MKL) 40
6.6.2 MKL Integration 41
6.7 Compiling Code for Parallel Execution – MPI Support 42
6.7.1 Compiling 42
6.7.2 OpenMPI 43
6.7.3 Platform MPI 43
6.7.4 Intel MPI 44
7 Debugging Code on HPC Wales 45
7.1 Debugging with idb 45
7.2 Debugging with Allinea DDT 45
7.2.1 Introduction 45
7.2.2 Command summary 46
7.2.3 Compiling an application for debugging 46
7.2.4 Starting DDT 46
7.2.5 Submitting a job through DDT 47
7.2.6 Debugging the program using DDT 51
8 Job Control 52
8.1 Job submission 52
8.1.1 Run Script 52
8.1.2 Submitting the Job 53
8.1.3 Resource Limits 54
8.2 Program Execution 54

8.3 Example Run Scripts 56
8.3.1 Example 1 56
8.3.2 Example 2 56
8.3.3 Example 3 56
8.3.4 Example 4 57
8.3.5 Example 5. Execution of the DLPOLY classic Code 58
8.3.6 Compiling and running OpenMP threaded applications 59
8.4 Job Monitoring and Control 61
8.4.1 The `bjobs` command 61
8.4.2 The `bpeek` command 63
8.4.3 The `bkill` command 64
8.4.4 The `bqueues` command 64
8.4.5 The `bacct` command 64
8.5 Interactive Jobs 65
8.5.1 Scheduling policies 65
8.5.2 Submitting Interactive Jobs 65
8.5.3 bsub -Is 66
8.5.4 Submit an interactive job and redirect streams to files 68
9 Using the SynfiniWay Framework 70
9.1 Access methods 70
9.1.1 SynfiniWay web interface 70
9.1.2 SynfiniWay Java Client 70
9.1.3 SynfiniWay user line commands 71
9.2 HPC Wales Portal 71
9.2.1 Entering the HPC Wales Portal 71
9.2.2 Opening a Gateway 73
9.3 SynfiniWay Gateway 75
9.3.1 Page layout 75
9.3.2 Tools: Run workflow 76
9.3.3 Tools: Monitor workflow 77
9.3.4 Tools: Global file explorer 77
9.3.5 Tools: Framework information 79
9.3.6 Tools: Preferences 80
9.3.7 Tools: Manuals 81
9.3.8 Leaving SynfiniWay 82

9.4 Using SynfiniWay Workflows 82
9.4.1 Introduction 82
9.4.2 Selecting workflow to use 82
9.4.3 Using workflow profiles 83
9.4.4 Defining workflow inputs 85
9.4.5 Submitting workflows 88
9.4.6 Track workflow state 89
9.4.7 Reviewing work files 91
9.4.8 Checking system information 92
9.4.9 Running application monitors 93
9.4.10 Stopping workflow 97
9.4.11 Cleaning workflows 98
9.5 Using the Data Explorer 99
9.5.1 Navigating file systems 99
9.5.2 Uploading files 100
9.5.3 Downloading files 102
9.5.4 Copying files and directories 103
9.5.5 Creating and deleting files and directories 105
9.5.6 Editing files 107
9.6 Developing Workflows 109
10 Appendix I. HPC Wales Sites 110
11 Appendix II. Intel Compiler Flags 112
12 Appendix III. Common Linux Commands 114
13 Appendix IV. HPC Wales Software Portfolio 116
13.1 Compilers 116
13.2 Languages 116
13.3 Libraries 117
13.4 Tools 125
13.5 Applications 131
13.6 Chemistry 131
13.7 Creative 133
13.8 Environment 134
13.9 Genomics (Life Sciences) 135
13.10 Benchmarks 143

Glossary of terms used in this document

API	Application Programming Interface
ARCCA	Advanced Research Computing @ Cardiff
CFS	Cluster File System
Cluster Management Node	A node providing infrastructural, management and administrative support to the cluster, e.g. resource management, job scheduling etc.
CMS	Cluster Management System
Compute Node	A node dedicated to batch computation
Condor	A project is to develop, implement, deploy, and evaluate mechanisms and policies that support High Throughput Computing (HTC) on large collections of distributively owned computing resources (see http://research.cs.wisc.edu/condor/)
Core	One or more cores contained within a processor package
CPU	Central Processing Unit
CYGWIN/X	Cygwin/X is a port of the X Window System to the Cygwin API layer (http://cygwin.com/) for the Microsoft Windows family of operating systems. Cygwin provides a UNIX-like API, thereby minimizing the amount of porting required.
DDN	Data Direct Networks, provider of high performance/capacity **storage** systems and processing solutions and services http://www.ddn.com/
DDR3	Double Data Rate Three
DIMM	Dual In-line Memory Module
ETERNUS	**ETERNUS** Storage Systems is a suite of storage hardware and software infrastructure. http://www.fujitsu.com/global/services/computing/storage/eternus/
FBDIMM	Fully Buffered DIMM
FileZilla	**FileZilla** is a cross-platform graphical FTP, FTPS and SFTP client with many features, supporting Windows, Linux, Mac OS X and more. http://download.cnet.com/FileZilla/3000-2160_4-10308966.html

GA	Global Array Toolkit from Pacific Northwest National Laboratory (http://www.emsl.pnl.gov/docs/global/)
Gb	Gigabit
GB	Gigabyte
Gbps	Gigabits Per Second
GCC	GNU Compiler Collection, as defined at http://gcc.gnu.org/
GFS	Global File System
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HPC	High Performance Computing
HPC Wales	High Performance Computing Wales (HPC Wales) is a £40million five-year project to give businesses and universities involved in commercially focussed research across Wales access to the most advanced and evolving computing technology available (http://www.hpcwales.co.uk/).
HPCC	HPC Challenge (Benchmark Suite)
HS	High Speed
HSM	Hierarchical Storage Management
HTC	High Throughput Computing (HTC). HTC systems process independent, sequential jobs that can be individually scheduled on many different computing resources across multiple administrative boundaries
IMB	Intel® MPI Benchmarks, Version 3.2.3
IPMI	Intelligent Platform Management Interface
kW	Kilowatt = 1000 Watts
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
Linux	Any variant of the Unix-type operating system originally created by Linus Torvalds
Login Node	A node providing user access services for the cluster

LSF	**L**oad **S**haring **F**acility from Platform Computing – the job scheduler on HPC Wales systems.
Lustre	A software distributed file system, generally used for large scale cluster computing
Memhog	A command that may be used to check **memory usage** in **Linux**.
Modules	Modules are predefined environmental settings which can be applied and removed dynamically. They are usually used to manage different versions of applications, by modifying shell variables such as PATH and MANPATH
MPI	Message Passing Interface - a protocol which allows many computers to work together on a single, parallel calculation, exchanging data via a network, and is widely used in parallel computing in HPC clusters.
MPI-IO	MPI-IO provides a portable, parallel I/O interface to parallel MPI programs
Multi-core CPU	A processor with 8, 12, 16 or more cores per socket.
NFS	Network File System
NIC	Network Interface Controller
Node	An individual computer unit of the system comprising a chassis, motherboard, processors and all additional components
OpenMP	An Application Program Interface (API) that may be used to explicitly direct ***multi-threaded, shared memory parallelism*** https://computing.llnl.gov/tutorials/openMP/
OS	Operating System
PCI	Peripheral Component Interconnect
PCM	Platform Cluster Manager (www.**platform**.com)
PCI-X	Peripheral Component Interconnect Extended
Portal	A web portal or links page is a web site that functions as a point of access to information in the Web
Processor	A single IC chip package (which may be single-core, dual-core, quad-core, hexa-core, octa-core, .. etc.)
PuTTY	An SSH and telnet client for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

QDR	Quad Data Rate (QDR) Infiniband (IB) delivers 40 Gbps per port (4x10 Gbps per lane)
RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RBAC	**Role-Based Access Control** is the mechanism for managing authorisation to objects managed by SynfiniWay, e.g., workflows, filesystem entry points.
RSS	**RSS** (originally RDF Site Summary is a family of web feed formats used to publish frequently updated works
SAN	Storage Area Network
SAS	Serial Attached SCSI
SATA	Serial Advanced Technology Attachment
SCP	Secure copy
Scientific Gateway	Science Gateways enable communities of users sharing a scientific goal to use grid resources through a common interface.
SCSI	Small Computer System Interface
Sharepoint	Microsoft SharePoint is a business collaboration platform that makes it easier for people to work together http://sharepoint.microsoft.com/en-us/product/capabilities/Pages/default.aspx
SIMD	Single Instruction, Multiple Data
SMP	Symmetric Multiprocessor
SSH	Secure Shell, a remote login program
Sub-System	One of the distributed HPC Clusters comprising the HPC Wales computing Infrastructure
SynfiniWay	SynfiniWay is an integrated grid or cloud framework for job execution on distributed and heterogeneous environments, within single dispersed organisations and between separate organisations.
TB	Terabyte
TCP	Transmission Control Protocol
TFlops	TeraFlops = 10^{12} FLOPS (FLoating point Operations Per Second)

UNIX	An operating system conforming to the Single UNIX® Specification as defined by the Open Group at http://www.unix.org/, and will also embrace those operating systems which can be described as Unix-like or Unix-type (or equivalent).
UPS	Uninterruptible Power Supply
WinSCP	WinSCP is a SFTP client and FTP client for Windows. Its main function is the secure file transfer between a local and a remote computer. http://winscp.net/eng/index.php
x86-64	A 64-bit microprocessor architecture
XMING	**Xming** is an implementation of the X Window System for Microsoft Windows operating systems, including Windows XP, Windows Server 2003, Windows Vista etc.

1 An Introduction to the User Guide

The HPC Wales High Performance Computing service provides a distributed parallel computing facility in support of research activity within the Welsh academic and industrial user community.

The service is comprised of a number of distributed HPC clusters, running the Red Hat Linux operating system. The present focus lies with the High Throughput Computing (HTC) cluster at Cardiff University, although this guide is intended to provide a generic document for using any of the HPC Wales sites. A list of the other sites and how to access them can be found in Appendix I of this guide. Note at the outset that this user Guide should be read in conjunction with a tutorial for getting started with the High Performance Computing cluster that can be downloaded from the HPC Wales portal.

The Guide is structured as follows. Following this introduction, Section 2 provides an overview of the HPC Wales System as it will finally appear, focusing on the collaborative working environment and the design and purpose of the portal, scientific gateways and the proposed workflow-driven usage model. With Fujitsu's middleware fabric – SynfiniWay – at the heart of this usage model, the overall system has been designed to remove from the user the need for a detailed understanding of the associated infrastructure and exactly how the components of that infrastructure interoperate. Suffice it to say that much of this final solution remains under development and will not be fully operational in the very near future. In the coming months this Guide will be extended to include all of these features, but in the short term the present version of the Guide is intended primarily for experienced users who need to understand how the system works and wish to access it using secure shell (SSH) and the ssh command.

Thus for the new user or those new to Linux and HPC, HPC Wales has provided an alternative access mechanism through SynfiniWay. Building on section 2, a general overview of SynfiniWay is provided in section 9 of this guide, with details of the associated access provided in the "SynfiniWay Quick Start Guide".

Section 3 describes the first steps in using the using the HPC Wales systems, with details of the support contacts and how to request an account to use the System. Gaining access to the component platforms, from both a Unix and Windows environment, is described, together with an outline of the available file transfer mechanisms, again from either a Unix or Windows environment.

Section 4 describes the Linux platforms that are currently available, with an outline of the configurations available at Cardiff – the HTC cluster – and the Tier-1 clusters at Aberystwyth, Bangor, and Glamorgan. An introduction on how to use these systems is given, with more detail on the access mechanisms and file transfer protocols.

Sections 5 to8 provides much of the detail required in developing, testing and executing end-user applications. Section 5 introduces aspects of the user environment, with a detailed description of the use of environment modules and the associated module commands. The variety of available compilers and scientific libraries are described in section 6, along with descriptions of the various MPI options – Intel MPI, Platform MPI and Open MPI – required in building parallel application software. Section 7 describes the techniques for debugging parallel software, with a focus on Intel's idb and the DDT debugger from Allinea.

Section 8 looks to provide all the background required to run jobs on the clusters, under control of Platform Computing's LSF. A variety of example run scripts are presented that hopefully cover all the likely run time requirements of the HPC Wales user community, together with descriptions of how to submit, monitor and control just how jobs are run on the system.

Finally, section 9 provides an overview of the capabilities of SynfiniWay, and describes the first instantiation of the Scientific Gateways – in genomics and chemistry – that will come to dominate the modality of usage for many of the HPC Wales community.

In addition to the glossary, a number of Appendices are included, including (i) A summary of the HPC Wales sites (Appendix I), (ii) A listing of the most used Intel compiler Flags (Appendix II), and (iii) A listing of the most common linux commands (Appendix III).

2 An Overview of the HPC Wales System

HPC Wales comprises a fully integrated and physically distributed HPC environment that provides access to any system from any location within the HPC Wales network. The design fully supports the distributed computing objectives of the HPC Wales project to enable and support the strategic Outreach activities.

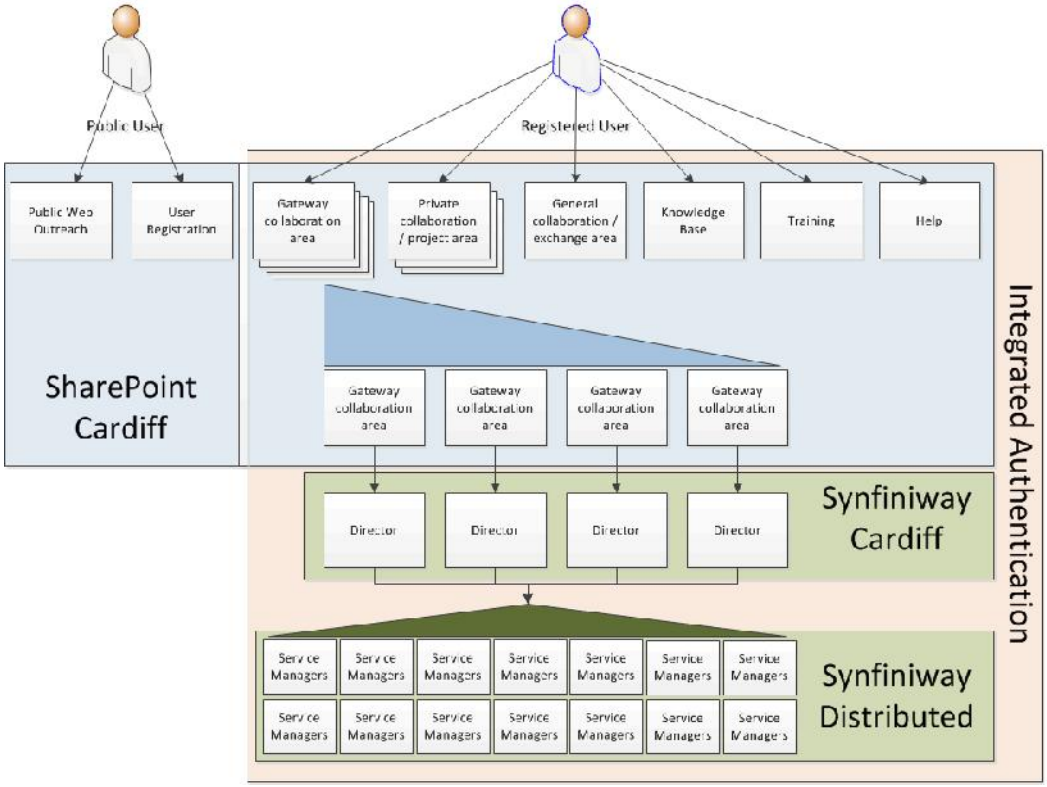


Figure 1.The logical hierarchy and integration of HPC Wales computing resources.

A SharePoint portal provides the public outreach web site and collaboration facilities including scientific gateways. Microsoft SharePoint is currently one of the leading horizontal portal products, social software products and content management products. The portal is still under active development, and will be built up over the coming months as the scientific gateways and other facilities are developed.

The Scientific gateways in SharePoint provide all the collaboration facilities for users of the gateways including, for example, content and people search, file sharing, ratings, forums, wikis, blogs, announcements, links. A wide range of web components will be available here, including RSS feed, RSS publisher, polls, blogs, wikis, forums, ratings, page note board/wall, tag cloud, picture libraries, picture library slideshow, shared documents, what's popular, site users, people browser, people search and refinement, announcements, relevant documents, charts and many others.

The integration of computing resources is delivered through the deployment of Fujitsu's proprietary work flow orchestration software system, SynfiniWay™ combined with Fujitsu's server clusters located at the two main hubs, Tier-1 and Tier-2 sites. This provides an integrated solution, enabling any user to access any system subject to a range of security and authorisation definitions.

The logical hierarchy of this interconnectivity is illustrated in Figure 1 above. SynfiniWay is a proven solution, supporting HPC global deployments, within large scale industrial organisations such as Airbus and Commissariat à l'énergie atomique.

The underlying hardware technology is based on Fujitsu's latest BX900 Blade technology. This technology is supported by Fujitsu's ETERNUS storage for filestore and backup together with a Concurrent File system from DDN, a recognized leader in this type of storage. Back-up and archiving is based on a combination of ETERNUS storage with Quantum tape library and Symantec back-up software.

This combination provides a robust and resilient solution which will enable HPC Wales to offer its capacity in a highly resilient format to external commercial and industrial users with the confidence of a large professional commercial data centre provider.

2.1 Collaborative working and User Productivity

SharePoint provides collaborative facilities for information sharing and distributed collaboration on projects. Documents may be shared and edited by multiple people in a controlled manner. Many facilities exist for sharing and gathering information such as forums, wikis, blogs, RSS, announcements, people and content search. Additionally users can receive email alerts when a change has been made or a new item has been added allowing them to keep up to date and in touch.

SynfiniWay allows users, located anywhere in the network, to access any of the computing systems, using data that may be located elsewhere in the network.

User access is controlled to provide the necessary levels of security and to control who can access what. User-access can be managed in a variety of ways including for example, on a thematic basis, so specific users can be restricted to a specific type of computing resource.

SynfiniWay makes it easy for geographically dispersed users to share data and work together. This can be extended to users in external organisations, in the UK or elsewhere, to encourage easy and rapid access to HPC resource

The HPC Wales System incorporates a broad range of capabilities to enable user productivity. A dedicated web environment facilitates knowledge exploration, information sharing and collaboration, using a single global identity and supported by single sign-on for easy navigation through the full-featured portal. HPC job execution is eased through a service-based approach to running applications, abstracting resources and networks, coupled with job workflow and global metascheduling for fully automated global execution. Removing the need for end-users to deal with the IT layer means their overall productivity is increased, less time is wasted on non-core activities leaving users more time to focus on their primary discipline science and research. In addition, the templates that are created through workflow encapsulation enable wider sharing and reuse of existing best practice HPC methods and processes.

2.2 The HPC Wales System Architecture

The HPC Wales System Architecture has two major components:

- User Environment

- Computing and Network Infrastructure.

The User Environment provides the user with access to the compute resources anywhere within the HPC Wales system through:

- User Portal
- Gateways
- SynfiniWay.

The Computing and Network Infrastructure provides:

- Front-end connectivity
- Computing resources
- Networking (within and between sites)
- File storage systems
- Backup and Archiving
- HPC Stack, management and operating system software
- Development software.

The User Environment and Computing and Network infrastructures are summarised below.

2.2.1 High Level Design

HPC Wales is implemented through a coherent and comprehensive software stack, designed to cover the wide-ranging needs HPC Wales today, and be both scalable and flexible enough to address future evolution and expansion.

- For most scientific and commercial end-user activity the first point of entry will be the HPC Wales Portal, an environment based on solid widely-deployed technology to provide a collaborative base for information exchange and learning. Pages within the portal will be mixed between public and secure access.
- Actual use of the HPC Wales sub-systems will mainly be channelled through dedicated web-based “gateways”, also accessed through a browser interface. Such gateways are the secure vehicle for consolidating knowledge in a particular application or business domain into a single point of access and sharing.
- In addition to the gateways, provision is made for the experienced users to access the systems directly using the secure shell protocol.

Beneath the user-facing layers, all of the HPC Wales sub-systems, network infrastructure and applications will be abstracted and virtualised using different middleware components – SynfiniWay, LSF, and other visualisation and monitoring tools.

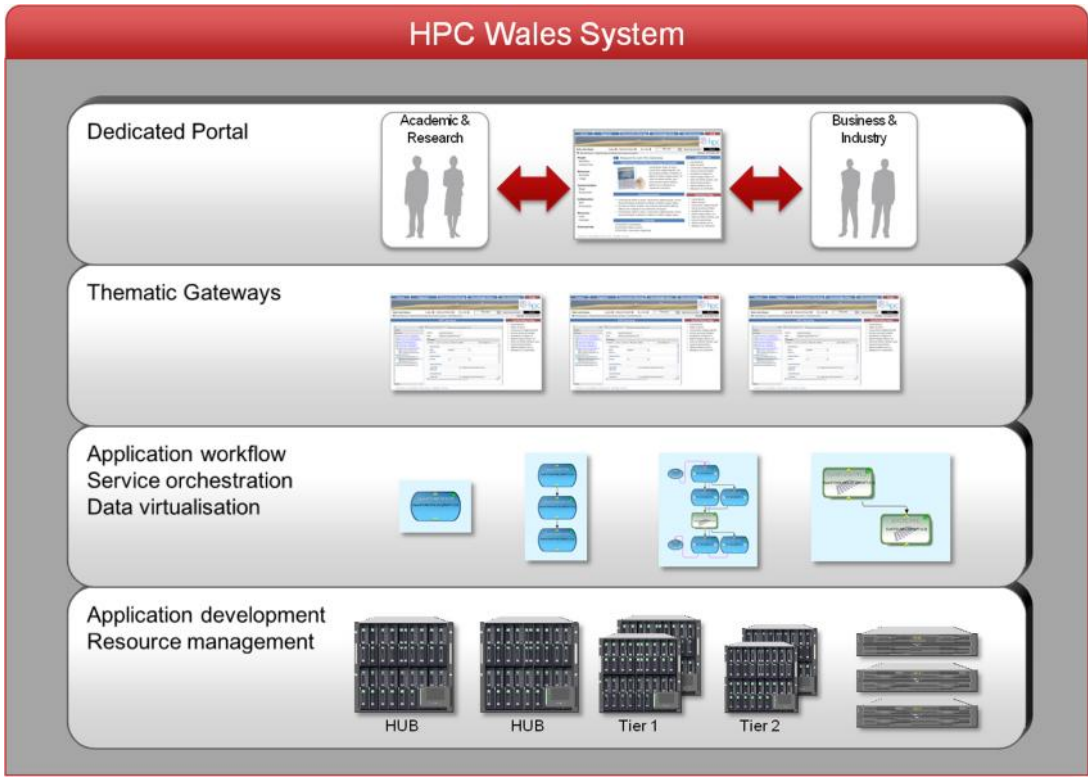


Figure 2. High level Design of the HPC Wales System

A unique approach in HPC Wales is to take full advantage of the inherent capabilities of SynfiniWay for removing the complexity usually associated with running HPC applications, and to present instead a generalised high-level service view to end-users. In this way scientists and researchers, new HPC users and commercial users, will be able to more easily utilise HPC to obtain insight into their areas of study.

Furthermore, such high-level services, representing optimal HPC methods and workflows, can become the assets of the provider – methods developer, research group, project team – capturing the intellectual capital of HPC Wales and releasing value-generation opportunities through the external user community.

2.3 System Software Environment and Usage Model

HPC Wales comprises a complete stack of software components to support the majority of the technical requirements. User activity of the HPC Wales System can divided into two broad categories:

- Application or method development – oriented towards interactive access to editing tools, profilers, debuggers.
- Application or method execution – emphasis on parameterisation and execution of work, movement of data, and analysis of output.

Development is supported primarily through interactive login to designated nodes, and the utilisation of the development environment provided by Intel and Allinea toolsets, and supported by the scientific and parallelisation libraries

Workflow development is provided through the inbuilt editor of SynfiniWay.

Execution, on local or remote systems, can be achieved through a command-line interface. As workflows are developed, the encapsulated applications may increasingly be submitted to the global HPC Wales System through a web interface.

The project to build a dedicated Portal and Gateway interface will be based on Microsoft SharePoint technology.

User management and web single sign-on will also be supported using Microsoft products.

Operating systems for the front-end service will use RedHat Linux, Windows Server and Windows HPC Server. Clusters will run CentOS Linux and Windows HPC Server. Dynamic switching between Linux and Windows will be enabled through the Adaptive Cluster component on specific sub-systems.

2.4 HPC Wales Computing and Networking Infrastructure

The HPC Wales hardware solution consists of HPC clusters, Storage, Networks, management servers, Visualisation systems and Backup and Archive distributed across the Hubs, Tier-1 and Tier-2 sites. There are 2 Hub sites one at Swansea[1] and the other at Cardiff. There are three Tier-1 sites at Aberystwyth, Bangor, and Glamorgan and Tier-2 sites at Swansea Metro & Trinity, Glyndwr at Technium Optic, Newport, UWIC, Springboard and Swansea.

The hub Tiers at Swansea and Cardiff have multiple compute sub-systems supported by common Front-End Infrastructure, Interconnect, Storage, and Backup and Archive. The Tier-1 and -2 sites follow the same architecture but with fewer components.

The HPC Wales systems are interconnected by a series of private links carried over the Public Sector Broadband Aggregation (PSBA) network. The two hubs are to be interconnected at 10Gb/Sec, whilst Tier-1 systems have 1Gb/Sec links and Tier-2 systems have 100Mb/Sec links. Some sites also benefit from 'Campus Connections' that provide direct links between the HPC Wales systems and host university networks. This negates the need to travel over the institution's Internet connection and thus provides substantially higher performance. Your local HPC Wales Technical representative will be able to provide more detail, or please contact the Support Desk.

[1] At the Dylan Thomas Centre

3 Using the HPC Wales Systems – First Steps

In order to make use of the service, you will first need an account. In order to obtain an account, please follow the instructions on the "Requesting an Account" section below. Although this guide assumes that you have an account have received your credentials, prospective users who do not as yet have an account will hopefully find the tutorials useful for developing an understanding of the how to use HPC Wales resources, although they will not be able to try out the hands-on examples.

Once your account has been created, we suggest you read sections 3 to 8 of this User Guide which describes how to log on to the systems, your user environment, programming for the cluster, and how to submit your jobs to the queue for execution.

3.1 Support

Please contact the support team if you have any problems, such as requiring the installation of a package or application.

The support desk email address is support@hpcwales.co.uk

The support desk phone number is 0845 2572207

3.2 Requesting an account

To request an account please contact your local campus representative. The following information is required to set up an account:

- Name
- Institution
- HPC Wales project
- Email address
- Contact telephone (optional)

Once the account is created in the Active Directory user management system it can be enabled to use the HPC Wales Portal, one or more Gateways and the SynfiniWay framework. This authorisation can be submitted to your campus representative at or after your initial account request. Within the Gateway and SynfiniWay system authorisation is controlled separately at various levels and with different roles.
Examples of the levels include:

- Gateway type or theme
- Project
- Application

Roles to use these levels are:

- Reader
- Contributor
- Owner

Your authorisation request should specify which entity you would like to use and the role you require. Permission for a given level will be referred to the nominated owner of that entity. Full details of the available entities and roles will be provided by your campus representative.

3.3 Accessing the HPC Wales Systems

Access to the system is via secure shell (SSH, a remote login program), through a head access node (or gateway), based in Cardiff which is available from the internet[2]. Wikipedia is a good source of information on SSH in general and provides information on the various clients available for your particular operating system.The detailed method of access will depend on whether you are seeking to connect from a Unix, Macintosh or Windows computer. Each case is described below.

3.3.1 Gaining Access from a UNIX environment

The head access node, or gateway, can be accessed using the ssh command.

```
ssh -X <username>@login.hpcwales.co.uk
```

where "**username**" should be replaced by your HPC Wales username. Note that the –X option enables X11 forwarding – this will be required if you are intending to run graphical applications on the HPC Wales login nodes and need to open a window on your local display.

For example, if your username (i.e. account id) was "`jones.steam`" then you would use:

```
ssh -X jones.steam@login.hpcwales.co.uk
```

You will be required to set a password on your first login. Details on password specification and how to change your password at a later date are given below in section 3.3.3. Note at this point that for reasons of security your password should contain a mix of lower and upper case and numbers.

Successfully logging into the HPC Wales login node will be accompanied by the following message,

```
------------------ HPC Wales Login Node ------------------------
Cyfrifiadura Perfformiad Uchel biau'r system gyfrifiadur hon.
Defnyddwyr trwyddedig yn unig sydd a hawl cysylltu a hi neu
fewngofnodi.  Os nad ydych yn siwr a oes hawl gennych, yna
DOES DIM HAWL. DATGYSYLLTU oddi wrth y system ar unwaith.
This computer system is operated by High Performance Computing
Wales.
Only authorised users are entitled to connect or login to it.
If you are not sure whether you are authorised, then you
ARE NOT and should DISCONNECT IMMEDIATELY.
--------------------Message of the Day------------------------------
Service Update 15/03/2013
Following the recent file system issues with the HTC cluster, the
service has now been resumed.
Please accept our sincere apologies for the outage.
```

[2] If you are based on an academic campus then it is possible you will be able to access one of the HPC Wales cluster login nodes directly, the technical team will be able to advise on the best way of connecting.

```
Service Update 17/03/2013

The Cardiff home directories are now mounted on the Access Nodes
once again.  Please contact support for access to any data stored
on the Access Nodes during the file system outage.
-------------------------------------------------------------------
You are now logged into the HPC Wales Network, please type
'hpcwhosts' to get a list of the site cluster login servers.
```

As indicated above, executing the "hpcwhosts" command provides a list of accessible systems, thus

```
$ hpcwhosts
HPC Wales Clusters Available

Location      Login Node(s)
------------------------------------------------------
Cardiff:      cf-log-001     cf-log-002     cf-log-003
Aberystwyth:  ab-log-001     ab-log-002
Bangor:       ba-log-001     ba-log-002
Glamorgan:    gl-log-001     gl-log-002
```

enabling you to ssh to any of the other site login servers (A full list of these servers can be found as Appendix I of this guide). Thus connecting to the Cardiff HTC system is achieved using ssh thus:

```
ssh cf-log-001
```

giving you access to a login node – part of the cluster reserved for interactive use (i.e. tasks such as compilation, job submission and control. This access will typically be accompanied by the following message.

```
Welcome to HPC Wales

This system is for authorised users, if you do not have authorised
access please disconnect immediately.

Password will expire in 8 days
Last login: Sat Jan 26 17:57:12 2013 from cf-log-102.hpcwales.local
==========================================================================
For all support queries, please contact our Service team on 0845 257
2207
or at support@hpcwales.co.uk

-------------------------Message of the Day----------------------

Happy New Year from HPC Wales!
==========================================================================
[username@log001 ~]$
```

3.3.2 Gaining Access from a Windows environment

If you are logging into your account from a Linux or Macintosh computer, you will have ssh available to you automatically. If you are using a Windows computer, you will need an SSH client such as PuTTY to access the HPC Wales systems.

PuTTY is a suitable freely downloadable client from

http://www.chiark.greenend.org.uk/~sgtatham/putty/.

If you are intending to use an application that uses a GUI i.e., if graphical applications running on the login nodes need to open a window on your local display, then it will be necessary to install an X-server program such as Xming or Cygwin/X on your Windows machine.

Having installed PuTTY and Xming on your PC, launch PuTTY and in the window enter the hostname as:

```
login.hpcwales.co.uk
```

and ensure the Port is set to 22 (the default). Optionally, under the SSH category, X11 section, set the "Enable X11 forwarding" if you are going to be using an X GUI. When you have logged into the HPC Wales access node, you will then be able to ssh to any of the other site login servers, a list of them can be found at the end of the guide.

When logging into **the HTC cluster** you will find yourself on one of the head nodes, which are the part of the cluster reserved for interactive use (i.e. tasks such as compilation, job submission and control).

3.3.3 Password Specification

You will be asked to change your password on the first login, and at regular intervals thereafter. Should you wish to change your password before the system requests you to, issue:

```
passwd
```

on a command line. You will be asked to type your existing password, and your new password twice. Your new password will need to contain at least one capital letter and a number and has a minimum length of eight characters.

3.4 File Transfer

The detailed method of access transferring files will depend on whether you are connecting from a Unix, Macintosh or Windows computer. Each case is described below.

3.4.1 Transferring Files from a UNIX environment

Files can be transferred between the cluster systems and your desktop computer using secure copy (SCP, remote file copy program): The syntax for scp is:

```
scp filename <username>@host:host_directory
```

Linux and Macintosh users will have these commands at their disposal already. So, again replacing <username> with your login name - `jones.steam` - then to copy a single file called "`data`" from your machine to the HPC system you would use:

```
scp data jones.steam@login.hpcwales.co.uk:
```

Where data would be transferred to the default `host_directory` – your home directory. Note that `scp` uses `ssh` for data transfer, and uses the same authentication and provides the same security as `ssh`.

OTHER EXAMPLES	
Command	`scp data1 jones.steam@login.hpcwales.co.uk:barny`
Description	Copy the file called "`data1`" into a directory on the HPC system called "`barny`".
Command	`scp -r data_dir jones.steam@login.hpcwales.co.uk:`
Description	Recursively copy (`-r` option) a directory called "`data_dir`" and all of its contents to your home directory.
Command	`scp -r data_dir jones.steam@login.hpcwales.co.uk:barny`
Description	Recursively copy a directory called "`data_dir`" and all of its contents into a directory called "`barny`" in the root of your HPCW filestore.

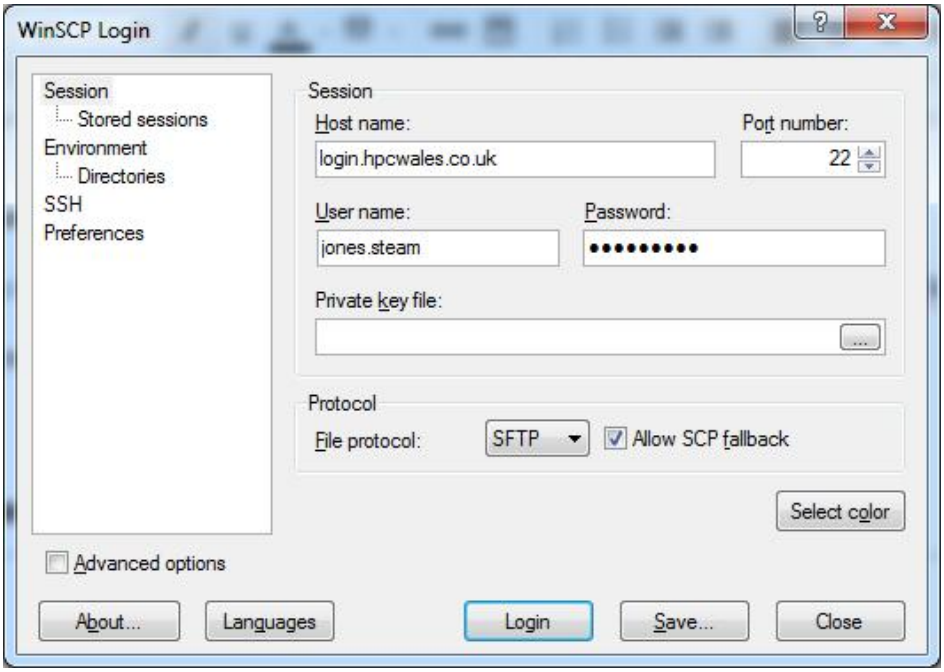
Note that if you are in a location with a campus connection then you will be able to copy files to your local site home directory.

3.4.2 Transferring Files from a WINDOWS environment

Windows users will need to install a suitable client, such as or FileZilla or WinSCP (from http://winscp.net/eng/index.php) which can be used to transfer files from Windows platforms.

Assuming WinSCP is the chosen client, once this is installed, you will receive a startup screen like that shown below in Figure 3 below, into which you must input your HPC Wales details.

You can use this interface to copy files to and from your PC to the HPC systems and back again. A full description of the use of WinSCP is beyond the scope of this document, however if you are used to a Windows explorer interface you may wish to use WinSCP in Explorer mode.



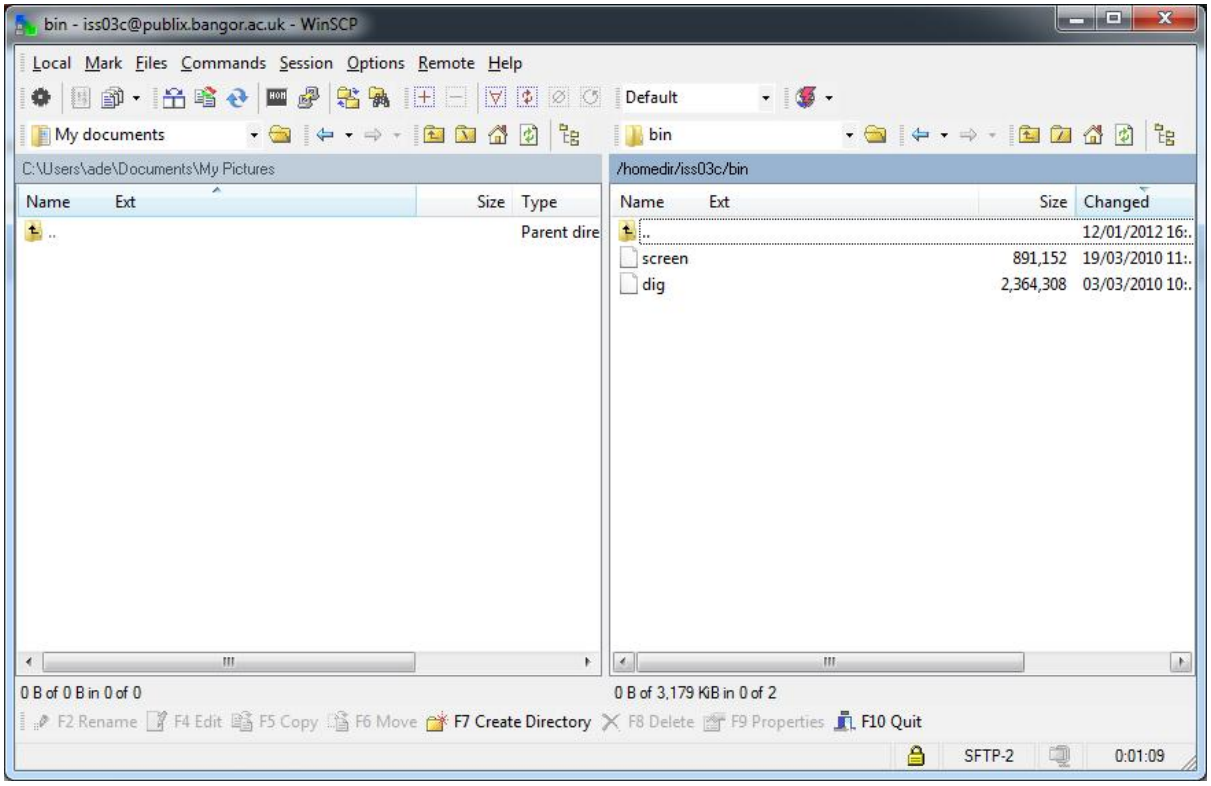

bin - iss03c@publix.bangor.ac.uk - WinSCP
Local Mark Files Commands Session Options Remote Help
Default
My documents
bin
C:\Users\ade\Documents\My Pictures
/homedir/iss03c/bin
Name Ext Size Type
Parent dire
Name Ext Size Changed
screen 891,152 19/03/2010 11:.
dig 2,364,308 03/03/2010 10:.
0 B of 0 B in 0 of 0
0 B of 3,179 KiB in 0 of 2
F2 Rename F4 Edit F5 Copy F6 Move F7 Create Directory F8 Delete F9 Properties F10 Quit
SFTP-2
0:01:09

Figure 3. Screenshots of the WinSCP dialogue boxes

4 The Cardiff Hub & Tier-1 Infrastructures

We briefly review the hardware infrastructure of the two HPC Wales sub-systems that are currently operational and supporting early HPC Wales projects – the Cardiff Hub and the Aberystwyth, Bangor, and Glamorgan Tier-1 systems

4.1 The Cardiff Infrastructure

The Cardiff infrastructure consists of Front-End infrastructure, Interconnect, Compute, Storage and Backup and archive. Each will be described here:

Front-end Infrastructure consists of Linux Management, Login and Install nodes, a Windows combined Head and Install node. Three nodes are designated for login usage and two of these also support the PCM, LSF and SynfiniWay functions. All servers are implemented by a similar PRIMERGY RX200 platform and are protected by a spare RX200 node. Each of these nodes is network installed. This enables the spare node to be quickly installed as any of the nodes it is protecting. There is a separate group of systems providing the SynfiniWay Directors and Portal.

Interconnect is provided via 1Gbit HPC Wales Network switches, 1Gbit Admin network switches, 10Gbit Internal Network, InfiniBand Internal Network and a Storage Area Network.

Compute resources are provided as specified:

- The Capacity and High Throughput Cluster (HTC) system.
- In the original Cardiff configuration, the Capacity system and HTC system were to share 162 nodes. Using the PCM Adaptive cluster module the 162 can be added to either the Capacity system or to the HTC system. The decision to upgrade the Capacity system to Intel's forthcoming Sandy Bridge technology means that the exact specification of the Capacity system is still to be determined i.e. this User Guide is specific at this stage to the HTC system.

Storage is provided by 2 systems; a high throughput low latency DDN storage system presenting via a Lustre file system and an ETERNUS DX400 system presented via a Symantec File System (SFS) cluster file system provides /home filestore space. The Symantec File System was formally known as the Veritas File System. It provides improved NFS performance and storage appliance like capabilities including data redundancy and migration. The DX400 system also provides other storage for Backup data deduplication storage, archive space, virtual machine native space and shared storage for the SynfiniWay/Portal environment.

Backup and Archive is provided by a Symantec NetBackup solution which provide scheduled backup to tape of local systems, remote systems are backed up over the network using a data deduplication technology to minimise the network data traffic. These deduplicated backups are staged to disk in the ETERNUS DX400 system. Off-site backups are created to tape as required and tapes are sent off site. NetBackup will also be used to archive stale data by regularly scanning the storage spaces for data that is inactive and archiving it firstly to the DX400 archive storage space. Following a further period of inactivity archive data will be moved to tape for long term archive.

Cluster management and operating system software - Cluster management is performed by the Platform Computing software stack that resides on the Cluster management nodes and controls the deployment of operating system images to the compute nodes. This software is also responsible for scheduling jobs to the individual clusters.

Development software is installed on the login nodes and is available to users. Dynamic libraries are installed on cluster nodes as part of the node image.

4.1.1 The Cardiff HTC Cluster

The HPC Wales HTC sub-system comprises a total of 167 nodes and associated infrastructure designed for High Throughput Computing, specifically:

- 162 BX922 dual-processor nodes, each having two six-core Intel Westmere Xeon X5650 2.67 GHz CPUs and 36GB of memory Westmere providing a total of 1994 Intel Xeon cores (with 3 GB of memory/core)
- 4 × RX600 X7550 dual processor Intel Nehalem nodes, 2.00 GHz, each with 128 GB RAM
- 1 × RX900 X7550 node with 8 Nehalem processors, 2.00 GHz and 512 GB RAM.
- Total memory capacity for the system of 6.85 TBytes.
- 100 TBytes of permanent storage.
- Interconnect - an InfiniBand non-blocking QDR network (1.2 µs & 40 Gbps).
- Lustre Concurrent File System (CFS, 200 TB storage), minimum data throughput of 3.5 GB/s.

	Server	Core	Mem [GB]	Disk [TB]	Count
Tier 1	BX922	12	36	0.292	162
Tier 2	RX600	16	128	4.000	4
Tier 3	RX900	64	512	2.400	1
Total		2,072	6,856	66	167

Intel Westmere processor

The Intel Westmere architecture includes the following features important to HPC:

- On-chip (integrated) memory controllers
- Two, three and four channel DDR3 SDRAM memory
- Intel QuickPath Interconnect (replaces legacy front side bus)
- Hyper-threading (reintroduced, but turned off for HPC computing)
- 64 KB L1 Cache/core (32KB L1 Data and 32KB L1 Instruction)
- 256KB L2 Cache/core
- 12MB L3 cache share with all cores (for HTC cluster 5650 processors)
- 2nd level TLB caching

The Intel Xeon 5650 (Westmere-EP) processors are employed in the Fujitsu BX922 blades. At 2.67GHz and 4FLOPS/clock period the peak performance per node is 12cores x 10.68GFLOPS/core = 128GFLOPS.

4.2 Filesystems

The HPC Wales HPC platforms have several different file systems with distinct storage characteristics. There are predefined, user-owned directories in these file systems for users to store their data. Of course, these file systems are shared with other users, so they are to be managed by either a quota limit, a purge policy (time-residency) limit, or a migration policy. Thus the three main file systems available for your use are:

Environment variable	*Location*	*Description*
$HOME	/home/$LOGNAME	Home filesystem
	/scratch/$LOGNAME	Lustre filesystem
	/tmp	local disk on each node for performing local I/O for duration of a job.

HPC Wales storage includes a 40GB SATA drive (20GB usable by user) on each node. $HOME directories are NSF mounted to all nodes and will be limited by quota. The scratch file system (/scratch) is also accessible from all nodes, and is a parallel file system supported by Lustre and 171TB of usable DataDirect Storage. Archival storage is not directly available from the login node, but will be accessible through scp.

The /scratch directory on HPC Wales Hub systems are Lustre file systems. They are designed for parallel and high performance data access from within applications. They have been configured to work well with MPI-IO, accessing data from many compute nodes.

Home directories use the NFS (network file systems) protocol and their file systems are designed for smaller and less intense data access – a place for storing executables and utilities. Use MPI-IO only on scratch filesystems.

To determine the amount of disk space used in a file system, cd to the directory of interest and execute the “df -k .” command, including the dot that represents the current directory. Without the dot all file systems are reported.

In the command output below, the file system name appears on the left (IP address, followed by the file system name), and the used and available space (-k, in units of 1 KBytes) appear in the middle columns, followed by the percent used, and the mount point:

```
$ df -k .
Filesystem           1K-blocks      Used Available Use% Mounted on
sfs.hpcwales.local:/vx/htc_home
                     80530636800 11506275328 68760342528  15% /home
```

To determine the amount of space occupied in a user-owned directory, cd to the directory and execute the du command with the -sh option (s=summary, h=units ’human readable):

```
$ du -sh
1.3G
```

To determine quota limits and usage on $HOME, execute the quota command without any options (from any directory). Note that at this point quota limits are not in effect.

```
$ quota
```

The major file systems available on HPC Wales are:

$HOME (**/home**)

- At login, the system automatically sets the current working directory to your home directory.
- Store your source code and build your executables here.
- This directory will have a quota limit (to be determined).
- This file system is backed up.
- The frontend nodes and any compute node can access this directory.
- Use $HOME to reference your home directory in scripts.

/scratch

- This directory will eventually have a quota limit (to be determined)
- Store large files here.
- Change to this directory in your batch scripts and run jobs in this file system.
- The scratch file system is approximately 171TB.
- This file system is not backed up.
- The frontend nodes and any compute node can access this directory.
- Purge Policy: Files with access times greater than 10 days will, at some point to be determined, be purged.
- NOTE: HPC Wales staff may delete files from /scratch if the file system becomes full, even if files are less than 10 days old. A full file system inhibits use of the file system for everyone. The use of programs or scripts to actively circumvent the file purge policy will not be tolerated.

/tmp

- This is a directory in a local disk on each node where you can store files and perform local I/O for the duration of a batch job.
- It is often more efficient to use and store files directly in /scratch (to avoid moving files from /tmp at the end of a batch job).
- The scratch file system is approximately 40 GB available to users.
- Files stored in the /tmp directory on each node must be removed immediately after the job terminates.
- Use /tmp to reference this file system in scripts.

$ARCHIVE

Future provision. Store permanent files here for archival storage. The HPC Wales policies for archival storage remain to be determined.

4.3 The Tier-1 Infrastructure at Aberystwyth, Bangor, and Glamorgan

The Tier-1 infrastructure consists of Front-End infrastructure, Interconnect, Compute, Storage and Backup and archive client software. Each will be described here:

Front-end Infrastructure consists of Linux Management, Login and Install nodes. Two nodes are designated for login usage and two of these also support the PCM, LSF and SynfiniWay functions. All servers are implemented using a similar PRIMERGY RX200

platform and are protected by a spare RX200 node. Each of these nodes is network installed. This enables the spare node to be installed quickly as any of the nodes it is protecting.

Interconnect is provided by QDR Infiniband switches within each of the BX900 blade chassis with interfaces provided for each of the blades. The BX900 hosted Infiniband switches are connected in a triangular topology with 9 QDR connections between each chassis pair. Additionally, 1Gbit HPC Wales Network switches, 1Gbit Admin network switches, together with a 10Gbit network providing access to file system storage.

Compute resources are provided as specified:

- Medium HPC system of 648 Westmere X5650 2.67GHz cores

Storage is provided by an ETERNUS DX80 system using the Symantec File System and presented as a NFS file system providing /home filestore space. The Symantec File System was formally known as the Veritas File System. It provides improved NFS performance and storage applicance like capabilities including data redundancy and migration.

Backup and Archive is provided by a Symantec NetBackup solution that provides scheduled backup, clients are backed up over the network using a data deduplication technology to minimise the network data traffic. These deduplicated backups are staged to disk in the hub ETERNUS DX400 system. Off-site backups are created to tape as required and tapes are sent off site. NetBackup will also be used to archive stale data by regularly scanning the storage spaces for data that is inactive and archiving it firstly to the DX400 archive storage space. Following a further period of inactivity archive data will be moved to tape for long term archive.

Cluster management and operating system software - Cluster management is performed by the Platform Computing software stack that resides on the Cluster management nodes and controls the deployment of operating system images to the compute nodes. This software is also responsible for scheduling jobs to the individual clusters.

Development software is installed on the login nodes and is available to users. Dynamic libraries are installed on cluster nodes as part of the node image.

4.4 An Introduction to Using the Linux Clusters

The clusters are running Linux, so you will need some familiarity with Linux shell commands.

In most cases you will need to write your own software to take advantage of the cluster, and you will need to write your code specifically to work in a parallel environment. As the clusters are shared, your program has to be submitted to a queue, where it will wait to be executed as soon as the computing resources are available. Because of this, you cannot interact with your code at run-time, so all input and output must be done without any user intervention.

In order to make best use of the HPC system, you will need to 'parallelize' your code. Your program must be broken down into a number of processes that will run on the compute nodes, and these processes need to communicate with each other in order to track progress and share data. The means of communication is implemented on the HPC Wales systems by a standard called the Message Passing Interface (MPI). MPI is a protocol which allows many computers to work together on a single, parallel calculation, exchanging data via a network, and is widely used in parallel computing in HPC clusters.

4.5 Accessing the Clusters

4.5.1 Logging In

The user is referred to section 3 where access through the HPC Wales gateway node has been described. Thus assuming access is from a Unix system, the head access node, or gateway, can be accessed using the ssh command, thus

```
$ ssh -X <username>@login.hpcwales.co.uk
```

Subsequent access to one of the login nodes at Cardiff, Aberystwyth, Bangor, and Glamorgan is then carried out simply by using

```
$ ssh cf-log-001
```

or

```
$ ssh gl-log-001
```

Respectively, in both cases seeking access to the first of the three login nodes available.

If you are based on an academic campus then it may be possible for you to access one of the HPC Wales cluster login nodes directly, albeit by IP address rather than login name.

Thus connecting directly to the Cardiff HTC system from the ARCCA Raven cluster is currently by IP address only. The IP addresses of the three HTC login nodes at Cardiff are:

194.83.32.1 (log-001)
194.83.32.2 (log-002)
194.83.32.3 (log-003)

The machine can thus be accessed using secure shell (SSH) and the ssh command, from for example the Raven login nodes, part of the ARCCA Facility, using the following ssh command:

```
$ ssh username@194.83.32.1
```

where "**username**" should be replaced by your HPC Wales username and the IP address can be any of those given above. This will give you access to a login node where you can submit jobs and compile applications. In similar fashion, direct access to the two login nodes at Glamorgan is by IP address, where the address of the two login nodes are as follows:

10.211.4.1 (log-001)
10.211.4.2 (log-002)

The Glamorgan cluster can thus be accessed using secure shell (SSH) and the ssh command, from for example the Raven login nodes, using the following ssh command:

```
$ ssh username@10.211.4.2
```

If the above approach is not operational in practice, please seek advice from the HPC Wales technical team to advise on the best way of connecting.

4.5.2 File Transfer

The user is referred to section 3.4 for advice on how best to transfer files between the cluster systems and your desktop computer with secure copy (SCP). Suffice it to say that transferring files directly can be accomplished using the appropriate IP address from those

given above e.g. to copy a single file called "`data`" from your machine to the Glamorgan login node – log-002 – would be accomplished thus:

```
$ scp data username@10.211.4.2:
```

5 The User Environment

5.1 Unix shell

The most important component of a user's environment is the login shell that interprets text on each interactive command line and statements in shell scripts. Each login has a line entry in the `/etc/passwd` file, and the last field contains the shell launched at login. To determine your login shell, use:

```
$ echo $SHELL
/bin/bash
```

You can use the `chsh` command to change your login shell. Full instructions are in the `chsh` man page. Available shells are defined by the `/etc/shells` file, along with their full-path.

To display the list of available shells with `chsh` and change your login shell to `tcsh`, execute the following:

```
$ chsh -l chsh -s /bin/tcsh
```

5.2 Environment variables

The next most important component of a user's environment is the set of environment variables. Many of the UNIX commands and tools, such as the compilers, debuggers, profilers, editors, and just about all applications that have GUIs (Graphical User Interfaces), look in the environment for variables that specify information they may need to access. To see the variables in your environment execute the command `env`:

```
$ env
HOME=/home/username
PATH=/app/intel_v11/Compiler/11.1/072/bin/intel64:
/app/libraries/impi/4.0.3.008/intel64/bin:
/opt/intel/impi/4.0.0.025/intel64/bin:
/opt/lsf/7.0/linux2.6-glibc2.3-x86_64/etc:
/opt/lsf/7.0/linux2.6-glibc2.3-x86_64/bin:/opt/kusu/bin:
/opt/kusu/sbin:/usr/kerberos/bin:
/opt/intel/clck/1.6:/usr/bin:/bin:/usr/sbin:/sbin:/usr/share/centri
fydc/bin:/usr/lib64:/home/username/bin:.
```

The variables are listed as keyword/value pairs separated by an equal (=) sign, as illustrated above by the `$HOME` and `$PATH` variables.

Notice that the `$PATH` environment variable consists of a colon (:) separated list of directories. Variables set in the environment (with `setenv` for C shells and export for Bourne shells) are carried to the environment of shell scripts and new shell invocations, while normal shell variables (created with the set command) are useful only in the present shell. Only environment variables are displayed by the `env` (or `printenv`) command. Execute `set` to see the (normal) shell variables.

5.3 Startup scripts

All UNIX systems set up a default environment and provide administrators and users with the ability to execute additional UNIX commands to alter the environment. These commands are sourced. That is, they are executed by your login shell, and the variables (both normal

and environmental), as well as aliases and functions, are included in the present environment.
Basic site environment variables and aliases are set in the following files:

`/etc/csh.cshrc` {C-type shells, non-login specific}
`/etc/csh.login` {C-type shells, specific to login}
`/etc/profile` {Bourne-type shells}

HPC Wales coordinates the environments on several systems. In order to efficiently maintain and create a common environment among these systems, the default shell in the HPC Wales environment is bash, with two files provided in your home directory:

```
.bashrc - Please do not change this.
.myenv – This is for your customisations.
```

If you have experience in altering your `.bashrc`, then please undertake these changes in `.myenv` as `.bashrc` can be overwritten by the system (`genconfig`).

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
# User specific aliases and functions
#####################################################
#HPC Wales Test and Dev + HTC system
#####################################################
export PATH=${PATH}:.
#####################################################
# User specific aliases and functions
#####################################################
. $HOME/.myenv
```

The `~/.bashrc` file is read or sourced on non-login interactive shells. We recommend strongly that Bash users place any module commands in their `~/.myenv` file – this will be sourced by the `~/.bashrc` file:

```
# .myenv
# User specific aliases and functions
# latest intel compilers,  mkl and Intel MPI
module load compiler/intel-11.1.072
# Intel MPI
module load mpi/intel-4.0.3.008
export PATH=$PATH:/usr/lib64:.
export OMP_NUM_THREADS=1
# Platform MPI
# module unload mpi/intel-4.0.3.008
# module load mpi/platform-8.1
```

5.4 Environment Modules

HPC Wales continually updates application packages, compilers, communications libraries, tools, and math libraries. To facilitate this task and to provide a uniform mechanism for accessing different revisions of software, HPC Wales uses the modules utility.

Environment Modules are predefined environmental settings which can be applied and removed dynamically. They are usually used to manage different versions of applications, by modifying shell variables such as PATH and MANPATH.

Modules are controlled through the **module** command. Common tasks are seeing which modules are available, loading and unloading modules. The following examples show each of these. Be aware that the modules available may differ from one sub-system to the next.

A summary of the module command options are shown in the Table 1 below:

`module avail`	Show available modules
`module load` *`modulename`*	Load a module
`module unload` *`modulename`*	Unload a module
`module swap` *`modulename`*	Swap a loaded module to a different version
`module purge`	Unload all modules
`module show`	Show the settings that a module will implement
`module help`	Display help information about the module command
`module help` *`modulename`*	Display help information about module *modulename*

Table 1: A summary of the module command options

Examples of the use of these command options are given in the examples below.

5.4.1 List Available Modules

The available module files can be determined using the "`module avail`" command. A complete list of all available modules on the Cardiff HTC system is given in section 13 (Appendix IV):

```
$ module avail
------------------------------ /app/modules/compilers ----------------------------
compiler/gnu                    compiler/gnu-4.6.2          compiler/intel-11.1
compiler/intel-12.0             compiler/portland/12.5      compiler/gnu-4.1.2
compiler/intel(default)         compiler/intel-11.1.072     compiler/intel-12.0.084
------------------------------ /app/modules/languages ----------------------------
Java/1.6.0_31(default)          R/2.14.1(default)           perl/5.14.2(default)
python/2.7.3                    R/2.13.1                    R/2.14.2
python/2.6.7(default)           python/2.7.3-gnu-4.6.2
------------------------------ /app/modules/libraries ----------------------------
```

```
GDAL/1.9.1                          hdf5/1.8.8-c++              mpi/platform-8.1
GDAL/1.9.1-gnu-4.6.2                hdf5/1.8.8-shared           mpi4py/1.3
GEOS/3.3.5                          hdf5/1.8.9-shared           mpiP/3.3
GEOS/3.3.5-gnu-4.6.2                hdf5/1.8.9-shared-gnu-4.6.2      muparser/2.2.2
Rmpi/0.6-1                          jasper/1.900.1(default)     ncl/2.1.17(default)
atlas/3.10.0-gnu-4.6.2(default)     lapack/3.4.2-gnu-6.2        netCDF/3.6.3(default)
beagle/1075(default)                libffi/3.0.11-gnu-4.6.2     netCDF/4.1.3
beamnrc/4-2.3.2                     libgeotiff/1.4.0            netCDF/4.1.3-shared
boost/boost_1_51_0                  libpng/1.2.50       netCDF/4.1.3-shared-gnu-
4.6.2cairo/1.12.0(default)          libpng/1.5.10(default)      nose/1.2.1-gnu-4.6.2
chroma/3.38.0                       libtool/2.4.2-gnu-4.1.2     numpy/1.6.2-gnu-4.6.2
delft3d/1301-gnu-4.6.2(default)     libunwind/1.0.1             octave/3.6.3-gnu-4.6.2
delft3d/5.00.10-gnu-4.6.2           libxml2/2.6.19              openssl/1.0.0g
egsnrc/4-2.3.2                      mcr/v717                    araFEM/2.0.819(default)
expat/2.0.1(default)                mpc/0.9                     parallel-netCDF/1.2.0
fftw/3.3(default)                   mpfr/3.1.0          pcre/8.32-gnu-4.6.2(default)
fftw/3.3-serial                     mpi/intel(default)          pdt/3.17(default)
fontconfig/2.8.0(default)           mpi/intel-4.0               petsc/3.1
freetype/2.4.9(default)             mpi/intel-4.0.3.008         petsc/3.2(default)
freetype/2.4.9-gnu                  mpi/intel-4.1               pixman/0.26.2(default)
ga/5.0.2(default)                   mpi/mpich2-1.5-gnu-4.6.2    proj/4.8.0
geant/4.9.5                         mpi/mpich2-1.5.1-gnu-4.1.2 pycogent/1.5.1(default)
gmp/5.0.2                           mpi/mpich2-1.5.1-gnu-4.6.2 qdp++/1.36.1
google-sparsehash/sparsehash-2.0.1/2.0.1       mpi/mvapich1  qmp/2.1.6
gsl/1.15(default)                   mpi/mvapich2                scipy/0.11.0-gnu-4.6.2
gts/120706(default)                 mpi/openmpi                 xerces/3.1.1
hdf5/1.8.6(default)                 mpi/openmpi-1.4.2           yasm/1.2.0(default)
hdf5/1.8.6-serial                   mpi/openmpi-1.5.4           zlib/1.2.7
hdf5/1.8.6-shared                   mpi/platform
--------------------------------- /app/modules/tools ------------------------------
allinea-ddt          git/1.7.2           moa/20120301  physica/2.11(default)      tau
antlr/2.7.7Q         gnuplot/4.6.0       mpitest       ploticus/2.41       texinfo/4.13
autoconf/2.68        impi_collective_tune nco/4.1.0    plotutils/2.6 weka/3.6.6(default)
automake/1.11.3      intel-inspector-xe  ncview/2.1.1  pvm3/3.4.6
cmake/2.8.7          ipm/0.983(default)  ne/2.4        scalasca/1.4(default)
ferret/6.72          mdtest/1.8.3(default)openss/2.0.1 sqlite/3071201
ffmpeg/1.0(default)  mencoder/1.1        paraview/3.14.0     subversion/1.7.2
--------------------------------- /app/modules/chemistry --------------------------
Gromacs/4.5.5-double       gamess/20110811(default)   nwchem/6.0-serial  vasp/5.2.12
Gromacs/4.5.5-single(default)    gamess-uk/8.0(default)     vasp/5.2(default)
crystal/9.10                     lammps/27Oct11(default)    vasp/5.2-debug
dlpoly-classic/1.8(default)      nwchem/6.0(default)        vasp/5.2-platform
--------------------------------- /app/modules/creative ---------------------------
Blender/2.47(default)            MentalRay-3.6-3DS/3.6.51(default)
MentalRay-3.10/3.10.1(default)   null
--------------------------------- /app/modules/financial --------------------------
null
--------------------------------- /app/modules/genomics ---------------------------
ABySS/1.2.7          R_Geneland/4.0.3    clustalw/2.1         muscle/3.8.31
AmberTools/12        R_MASS/7.3-16       clustalw-mpi/0.13    openbugs/3.2.1
AmpliconNoise/1.25   R_ade4/1.4-17       dialign/2.2.1        pauprat/03Feb2011
BEAST/1.7.1          R_adegenet/1.3-3    dialign-tx/1.0.2     plink/1.07
BLAST/2.2.25         R_ape/2.8           eigenstrat/3.0       prank/100802
BLAST+/2.2.25        R_gee/4.13-17       eigenstrat/4.2       pynast/1.1
BWA/0.5.9            R_pegas/0.4         fasttree/2.1.3       qiime/1.3.0
BioPerl/1.6.1        R_seqinr/3.0-6      impute/2.1.2         raxml/7.2.8
CABOG/6.1            R_spider/1.1-1      lastz/1.02.00        rdp_classifier/2.2
Curves/1.3           SAMtools/3.5        mach/1.0.17          rmblast/1.2
GATKSuite/1.1.23     SHRiMP/2.2.0        mach2dat/1.0.19      transalign/1.2
GeneMarkS/4.6b       SOAP2/2.21          mafft/6.864          trf/4.0.4
HMMER/3.0            Spines/1.15         maq/0.7.1            uclust/1.2.22
JAGS/3.2.0           T-Coffee/9.02.r1228 molquest/2.3.3       velvet/1.1.06
```

```
Kalign/2.03         VMD/1.9.1           mpiBLAST/1.6.0
LAGAN/2.0           bowtie/0.12.7       mrbayes/3.2.0
R_Biostrings/2.22.0 bowtie2/2.0.2       muscle/3.3
-------------------------------- /app/modules/materials --------------------------
null
-------------------------------- /app/modules/environment ------------------------
DELFT3D/4.00/4.00.01      TELEMAC/v6p1 wps/3.4(default)    wrfda/3.4.1(default)
Gerris/121021(default)    TELEMAC/v6p1_impi_4.0.3.008      wps/3.4_platform-mpi
ROMS/20110311(default)    ncl-ncarg/6.1.0-beta(default)    wrf/3.4(default)
SWAN/40.85(default)       pism/0.4(default)                wrf/3.4_platform-mpi
-------------------------------- /app/modules/benchmarks -------------------------
imb/3.2(default)    iozone/modulefile   mpi_nxnlatbw/0.0    stream/5.6(default)
-------------------------------- /app/modules/system -----------------------------
cpu/auto(default)         dot           interconnect/Ethernet      use.own
cpu/sandybridge           http-proxy    interconnect/infiniband
cpu/westmere              interconnect/auto(default) null
```

5.4.2 Show Module Information

The "**`module show`**" command gives details about what changes to the user environment will be caused by loading the module:

```
$ module show Gromacs/4.5.5-double
-------------------------------------------------------------
/app/modulefiles/Gromacs/4.5.5-double:
module-whatis    Gromacs 4.5.5-double
prepend-path     PATH /app/chemistry/Gromacs/4.5.5-double/bin
prepend-path LD_LIBRARY_PATH /app/chemistry/Gromacs/4.5.5-double/lib
-------------------------------------------------------------
```

5.4.3 Loading Modules

The "`module load`" command is used to load settings for a specific module, for example:

```
$ module load compiler/intel-11.1
```

If you do not specify a specific version then the default is chosen and so "module load compiler" is equivalent to "`module load compiler/intel`". It is also possible to load multiple modules in a single "`module load`" command, thus

```
$ module load compiler/gnu mpi/intel
```

5.4.4 Unloading Modules

The "`module unload`" command is used to remove a module, for example:

```
$ module unload compiler/gnu
```

There may be dependencies, prerequisites and conflicts between modules. A module may refuse to load if there is a conflicting module already loaded and a module may attempt to load other modules which it requires.

For example loading an MPI module when there is not a compiler module loaded will automatically load the default compiler module:

```
$ module list
No Modulefiles Currently Loaded.
$ module load mpi/platform
$ module list
Currently Loaded Modulefiles:
  1) compiler/intel   2) mpi/platform
```

5.4.5 Verify Currently Loaded Modules

A list of currently loaded modules can be displayed through the "`module list`" command

```
$ module list
Currently Loaded Modulefiles:
  1) compiler/intel-11.1.072   2) mpi/intel-4.0.3.008
```

5.4.6 Compatibility of Modules

Most applications have been built with a specific compiler and, if required, MPI library. If the modules currently loaded are not compatible with the specific version of the application then the module will fail to load:

```
$ module load Gromacs
ERROR: This build of Gromacs is not supported with the currently loaded
MPI module.
```

For this reason it is advisable to use "**module purge**" before loading an application module:

```
$ module purge
$ module list
No Modulefiles Currently Loaded.
$ module load Gromacs
$ module list
Currently Loaded Modulefiles:
1) compiler/intel-11.1 2) mpi/intel-4.0 3) Gromacs/4.5.5-single
```

5.4.7 Other Module Commands

Some other useful module commands are illustrated below:

The "module help" command for a module shows some useful information about the application:

```
$ module help Gromacs/4.5.5-double
--- Module Specific Help for 'Gromacs/4.5.5-double' ---
Loads PATH and LD_LIBRARY_PATH settings for Gromacs 4.5.5-double compiled
with intel-11.1 compiler and intel-4.0 MPI
An   example   input   file   and   job   script   are   available   in
```

```
/app/chemistry/Gromacs/4.5.5-double/example
```

The “module swap” command can be used to change between different versions of a module:

```
$ module list
Currently Loaded Modulefiles:
 1) compiler/intel-11.1  2) mpi/intel-4.0  3) Gromacs/4.5.5-single
$ module swap Gromacs/4.5.5-double
$ module list
Currently Loaded Modulefiles:
 1) compiler/intel-11.1  2) mpi/intel-4.0  3) Gromacs/4.5.5-double
```

6 Compiling Code on HPC Wales

If you are not using an existing software package, then it is likely you will need to compile the code yourself. Don't forget the support team are available to help if needs be. This section provides information for those who need to compile their own code in C, C++ or Fortran.

6.1 Details of the available compilers and how to use them

There are two C/C++/Fortran compiler suites available on HPC Wales: the Intel Cluster Toolkit and the Gnu Compiler Collection (GCC).

Language	GCC	Intel
C	gcc	icc
C++	g++	icpc
F77	gfortran	ifort
F90	gfortran	ifort
F95	gfortran	ifort

***Table 2**. Compilers names for the compiler suites on HPC Wales*

6.2 GNU Compiler Collection

6.2.1 Documentation

The GNU Compiler Collection website at http://gcc.gnu.org/ contains FAQs and further literature in addition to the compiler documentation linked to below. We generally advise compiling with the Intel compilers but for some software use of the GCC compilers is essential.

6.2.1.1 Version 4.1.2

Version 4.1.2 is installed on HPC Wales sub-systems. Documentation is available at:

- GNU C/C++ Compiler 4.1.2 Documentation

  http://gcc.gnu.org/onlinedocs/gcc-4.1.2/gcc/.
- GNU Fortran Compiler 4.1.2 Documentation

  http://gcc.gnu.org/onlinedocs/gcc-4.1.2/gfortran/.

```
$ module load compiler/gnu
```

6.3 Intel Compilers

The Intel compilers are heavily optimised for Intel's range of processors, and so are generally the best choice for software development on HPC Wales clusters. The compiler suite provides C/C++ and Fortran 77/90/95 compilers, highly optimised threaded math routines in the Math Kernel Library, and a debugger.

On **HPC Wales** you must use the module command to make the Intel compiler available to your current environment. If you just wish to use the Intel compilers you can simply load

```
$ module load compiler/intel
```

However if you also want to link to the Intel Maths Kernel Libraries (MKL), use the Profiling tools (ITAC) or Intel MPI (IMPI) it is simpler to load the meta-module

```
$ module load compiler/intel
```

This will load the recommended version of all the Intel Cluster Toolkit modules. Links to Intel compiler documentation can be found as follows:

Intel C++ compiler for Linux Knowledge Base

http://software.intel.com/en-us/articles/intel-c-compiler-for-linux-kb/all/1/

Intel Fortran compiler for Linux Knowledge Base

http://software.intel.com/en-us/articles/intel-fortran-compiler-for-linux-kb/all/1/

Intel Math Library Knowledge Base

and you'll also find `man` pages for each compiler command. The following common Intel compiler flags may be useful

Option	Description
-O	Default optimization (equivalent to -O2)
-O3	More aggressive optimization
-ipo	Enable interprocedural optimization
-xHost	Tune executable for current host architecture
-no-prec-div	Use faster but less precise floating point divide (Not IEEE compliant)
-ftz	Flush denormal results to zero
-assume buffered_io	(Fortran only) Buffer I/O writes for improved performance
-openmp	Enable OpenMP thread parallelism
-g	Generate debugging information
-check uninit	(Fortran only) Check for uninitialized variables at runtime (may affect performance so use for debugging only)

***Table** 3. Common Intel Compiler Flags*

6.4 Compiling Code – a Simple Example

Below is a very simple worked example of compilation using the Intel C Compiler. First make sure the Intel module is loaded:

```
module load compiler/intel
```

The next stage is to write a little code to compile, below is a very simple command, open an editor such as Vi or EMACS:

```
#include
main()
{
      printf("Hello, world\n");
}
```

Save this C code as `hello.c` and exit the editor. The next stage is to compile the code, common compiler flags and commands are listed in Table 3.

```
$ icc hello.c -o hello
```

icc being the compiler, `hello.c` being the filename to compile and -o `hello` meaning output a binary called `hello`. To test the application has compiled correctly, run it:

```
$ ./hello
```

And you should see the output:

```
Hello, world
```

More compiler options and information can be run with the `icc -help` and `man icc` commands.

6.5 Libraries

Some of the more useful load flags/options are listed below. For a more comprehensive list, consult the `ld` man page.

- Use the `-l` loader option to link in a library at load time: e.g. `ifort prog.f90 -l<name>`
- This links in either the shared library `libname.so` (default) or the static library `libname.a`, provided it can be found in ldd's library search path or the LD_LIBRARY_PATH environment variable paths.
- To explicitly include a library directory, use the `-L` option, e.g. `ifort prog.f -L/mydirectory/lib -l<name>`

In the above examples, the user's `libname.a` library is not in the default search path, so the L option is specified to point to the `libname.a` directory. (Only the library name is supplied in the -l argument - remove the lib prefix and the .a suffix.)

Many of the modules for applications and libraries, such as the `mkl` library module provide environment variables for compiling and linking commands. Execute the module help `module_name` command for a description, listing and use cases for the assigned environment variables. You can view the full path of the dynamic libraries inserted into your binary with the `ldd` command. The example below shows a listing for the `cpmd.x` binary:

```
[username@log002 SOURCE]$ ldd cpmd.x
libdl.so.2 => /lib64/libdl.so.2 (0x00002aaaaacc8000)
libmkl_core.so =>
/app/intel_v11/mkl/10.2.5.035/lib/em64t/libmkl_core.so
```

```
(0x00002aaaaaecc000)
libmkl_intel_lp64.so =>
/app/intel_v11/mkl/10.2.5.035/lib/em64t/libmkl_intel_lp64.so
(0x00002aaaab27f000)
libmkl_sequential.so =>
/app/intel_v11/mkl/10.2.5.035/lib/em64t/libmkl_sequential.so
(0x00002aaaab67a000)
libguide.so =>
/app/intel_v11/Compiler/11.1/072/lib/intel64/libguide.so
(0x00002aaaabe8e000)
        libpthread.so.0 => /lib64/libpthread.so.0
(0x00002aaaac036000)
libmpi.so.4 =>
/app/libraries/impi/4.0.3.008/intel64/lib/libmpi.so.4
(0x00002aaaac252000)
libmpigf.so.4 =>
/app/libraries/impi/4.0.3.008/intel64/lib/libmpigf.so.4
(0x00002aaaac71c000)
librt.so.1 => /lib64/librt.so.1 (0x00002aaaac793000)
libmpi_dbg.so.4 =>
/app/libraries/impi/4.0.3.008/intel64/lib/libmpi_dbg.so.4
(0x00002aaaaca54000)
libm.so.6 => /lib64/libm.so.6 (0x00002aaaace6c000)
libc.so.6 => /lib64/libc.so.6 (0x00002aaaad0ef000)
libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00002aaaad447000)
/lib64/ld-linux-x86-64.so.2 (0x00002aaaaaaab000)
```

A load map, which shows the library module for each static routine and a cross reference (`--cref`) can be used to validate which libraries are being used in your program. The following example shows that the `ddot` function used in the `mdot.f90` code comes from the MKL library:

6.6 Performance Libraries

ISPs (Independent Software Providers) and HPC vendors provide high performance math libraries that are tuned for specific architectures. Many applications depend on these libraries for optimal performance. Intel has developed performance libraries for most of the common math functions and routines (linear algebra, transformations, transcendental, sorting, etc.) for the EM64T architectures. Details of the Intel libraries and specific loader/linker options are given below.

6.6.1 Math Kernel Library (MKL)

The Math Kernel Library consists of functions with Fortran, C, and C++ interfaces for the following computational areas:

- BLAS (vector-vector, matrix-vector, matrix-matrix operations) and extended BLAS for sparse computations
- LAPACK for linear algebraic equation solvers and eigensystem analysis
- Fast Fourier Transforms

- Transcendental Functions

In addition, MKL also offers a set of functions collectively known as VML -- the Vector Math Library VML is a set of vectorized transcendental functions that offer both high performance and excellent accuracy compared to the libm functions (for most of the Intel architectures). The vectorized functions are considerably faster than standard library functions for vectors longer than a few elements.

To use MKL and VML, first load the MKL module using the command `module load mkl`. Below is an example command for compiling and linking a program that contains calls to BLAS functions (in MKL). Note that the library is for use in a single node, hence it can be used by both serial compilers or by MPI wrapper scripts.

The following C and Fortran examples illustrate the use for the mkl library after loading the compiler module: module load compiler:

```
$ mpiicc myprog.c -L/app/intel_v11/mkl/10.2.5.035/lib/em64t \
    -Wl,--start-group \
    -lmkl_intel_lp64 -lmkl_sequential -lmkl_core \
    -Wl,--end-group -lpthread

$ mpiifort myprog.f90 -L/app/intel_v11/mkl/10.2.5.035/lib/em64t \
    -Wl,--start-group \
    -lmkl_intel_lp64 -lmkl_sequential -lmkl_core \
    -Wl,--end-group -lpthread
```

Notice the use of the linker commands `--start-group` and `--end-group`, which are used to resolve dependencies between the libraries enclosed within. This useful option avoids having to find the correct linking order of the libraries and, in cases of circular dependencies, having to include a library more than once in a single link line.

Assistance in constructing the MKL linker options is provided by the *MKL Link Line Advisor* utility (see below).

6.6.2 MKL Integration

When using the optimised Intel MKL libraries, in order to simplify selecting the appropriate libraries the Intel compilers support a flag which tells the compiler to work out which libraries to compile and link against. For example:

```
$ icc -mkl <other arguments>
```

Linking against MKL in particular can prove difficult for users new to the Intel Toolkit. It is recommended that you use the Intel MKL Link Advisor to create your link lines - at least as a starting point. Table 4 shows common values for the Link advisor:

Field	Recommend values
Select OS:	Linux
Select processor architecture:	Intel 64
Select compiler:	Intel | GNU C | GNU Fortran
Select dynamic or static linking	dynamic | Static

Select your integers length:	32-bit (lp64)
Select sequential or multi-threaded	sequential | multi-threaded
Select OpenMP library:	iomp5 *
Select Cluster library:	Scalapack *
Select MPI library:	Intel MPI | Open MPI *

***Table 4**. Useful starting values for Intel MKL Link advisor for HPC.*

You may not need to enter any values for the last three fields. Once you have entered your values the link advisor will provide you with a list of linker options e.g.

```
-L$MKLPATH $MKLPATH/libmkl_solver_lp64_sequential.a \
-Wl,--start-group -lmkl_intel_lp64 -lmkl_sequential -lmkl_core \
-Wl,--end-group -lpthread
```

that can be used directly if you are linking from the command line, for example

```
> ifort -L$MKLPATH $MKLPATH/libmkl_solver_lp64_sequential.a \
-Wl,--start-group -lmkl_intel_lp64 -lmkl_sequential -lmkl_core \
-Wl,--end-group -lpthread mycode.f90 -o mycode.3
```

If instead you are using a Makefile (rather than linking directly at the command line) you will need to add the above linker options to the LDFLAGS, CFLAGS or FFLAGS variable in your Makefile. You will also need to change references to $MKLPATH to $(MKLPATH), so that the make command understands them.

6.7 Compiling Code for Parallel Execution – MPI Support

6.7.1 Compiling

Instead of using the compiler commands directly, you should use the MPI wrappers which will take care of compiling your code and linking to the MPI libraries. The command options supplied to the MPI wrappers will be passed to the compiler and/or linker, so you can treat the wrappers as direct replacements for the compiler commands.

There are three distinct flavours of MPI available on HPC Wales sub-systems:

1. Open MPI + Intel Compilers
2. Platform MPI + Intel Compilers
3. Intel MPI + Intel Compilers (Recommend for performance)

Note that the associated GNU MPI wrapper scripts and libraries can be made available on request.

Language	Open MPI + intel	Platform MPI + intel	Intel MPI + Intel
C	mpicc	mpicc	mpiicc
C++	mpicxx	mpicxx	mpicpc
F77	mpif77	mpif77	mpiifort
F90	mpif90	mpif90	mpiifort

F95	mpif90	mpif90	mpiifort

Table 5. Names of MPI wrapper scripts for the different MPI implementations.

The wrapper commands are given in Table 5. These include

1. `mpicc, mpiCC`, for icc (C), and icpc (C++),
2. `mpif77`, f77 (Fortran 77),
3. `mpif90`, f90 (Fortran 90)

Here's an example which compiles a C program *prog.c*, producing the executable *prog*.

```
$ mpicc -o prog prog.c
```

6.7.2 OpenMPI

To load the related Intel compiler module, wrapper scripts and libraries use

```
$ module load mpi/openmpi
```

or

```
$ module load mpi/openmpi-1.4.2
```

or

```
$ module load mpi/openmpi-1.5.4
```

This will add the OpenMPI Version 1.4.2 bin, man and lib directory to PATH, MANPATH and LD_LIBRARY_PATH respectively. This will also load the related compiler module (if it is not already loaded) i.e. the Intel Compiler module. Note that the associated GNU MPI wrapper scripts and libraries can be made available on request.

6.7.3 Platform MPI

To load the related Intel compiler module, wrapper scripts and libraries use

```
$ module load mpi/platform
```

or

```
$ module load mpi/platform-8.1
```

This will add the PlatformMPI Version 8.1 bin, man and lib directory to PATH, MANPATH and LD_LIBRARY_PATH respectively. This will also load the related compiler module (if it is not already loaded) i.e. the Intel Compiler module. Note that the associated GNU MPI wrapper scripts and libraries can be made available on request.

6.7.4 Intel MPI

When you load Intel MPI it will add the Intel compilers to your path. Note that the wrapper scripts for the GNU compiler have different names - unlike the OpenMPI wrapper script which all have identical names and therefore require their own module).

```
$ module load mpi/intel
```

or

```
$ module load mpi/intel-4.0
```

or

```
$ module load mpi/intel-4.0.3.008
```

or

```
$ module load mpi/intel-4.1
```

or

```
$ module load mpi/ 4.1.0.030
```

7 Debugging Code on HPC Wales

7.1 Debugging with idb

The Intel compiler suite includes a debugger, idb. This can be added to your environment with the module command to load the appropriate idb settings.

In order to make full use of the debugger, it's necessary to compile your code with symbol table information; at the simplest level, this can be done by using the **-g** flag for the C/C++ or Fortran compiler. See the man page for more details of the available debugging options.

As debugging parallel applications can be a major challenge, you should first make sure that your job runs properly as a serial application, and debug as far as possible in a normal environment.

Below is a short example showing a C program being compiled with symbolic debugging info, then the resulting executable being loaded into idb. A breakpoint is set at main(), the program is executed, and after stopping at main(), is single-stepped twice. Typing quit exits idb.

```
$ mpicc -g -debug extended -o prog prog.c
$ idb prog
Intel(R) Debugger for Itanium(R) -based Applications, Version 9.1-
25, Build 20060928
------------------
object file name: debug
Reading symbolic information from /home/zzlg/code/prog...done
(idb) stop in main
[#1: stop in int main(int, char**)]
(idb)
[1] stopped at [int main(int, char**):19:16 0x400000000000bc01]
     19   unsigned int delay = 0;
(idb)
(idb) step
stopped at [int main(int, char**):23:28 0x400000000000bc11]
     23   int rank, procs, source, tag = 0;
(idb)
(idb) step
stopped at [int main(int, char**):27:13 0x400000000000bc21]
     27   MPI_Init (&argc, &argv);      /* starts MPI */
(idb)
(idb) quit>
```

7.2 Debugging with Allinea DDT

7.2.1 Introduction

Allinea DDT, the Distributed Debugging Tool, is a graphical debugger for parallel codes. It is capable of debugging serial programs; multi-process MPI codes to high process counts;

multi-threaded programs with OpenMP or Pthreads and mixed-mode multi-process (MPI) / multi-thread (OpenMP) programs. C, C++, and Fortran are all supported by DDT.

DDT is integrated with the LSF queuing system on all HPC Wales machines, allowing users to submit jobs through the DDT GUI. The current version available is 3.0.

This document describes the basic steps to get prepare a code for debugging, how to start the Allinea DDT GUI and how to submit a job through DDT. See the DDT User Guide - `/app/doc/tools/allinea/ddt/usermanual.pdf` for instructions on debugging the program once submitted.

7.2.2 Command summary

```
$ ssh –CX jim.enright@10.211.1.129 # login with X-forwarding
$ module load compiler/intel # load the compiler
$ module load mpi/intel # load mpi library
$ mpicc –g testDebug.c –o testDebug.x # compile program with debugging enabled and default optimisation
$ module load allinea/ddt # load the allinea ddt module
$ ddt &  # start ddt
```

7.2.3 Compiling an application for debugging

- Load the compiler and mpi library as per usual with the module commands. For example, to load the intel compiler and mpi library:

```
$ module load compiler/intel mpi/intel
```

- Code should be compiled with the –g flag to generate debugging information for use by DDT. This will allow source lines to be displayed with DDT.

```
$ mpicc -g testDebug.c -o testDebug.x
```

- It is also recommended, though not strictly necessary, to turn down the compiler optimisation level as some optimisations can make the debugged source difficult to follow.

7.2.4 Starting DDT

- To run the DDT GUI an X-server (e.g. Xming or Cygwin/X) needs to be running on your local machine. You can then login using ssh –CX <username>@<machineName> to forward the DDT GUI to your local machine.
- Load the Allinea DDT module with the command

```
$ module load allinea-ddt
```

- Start the DDT GUI:

```
$ ddt
```

This will bring up the DDT Welcome Screen:

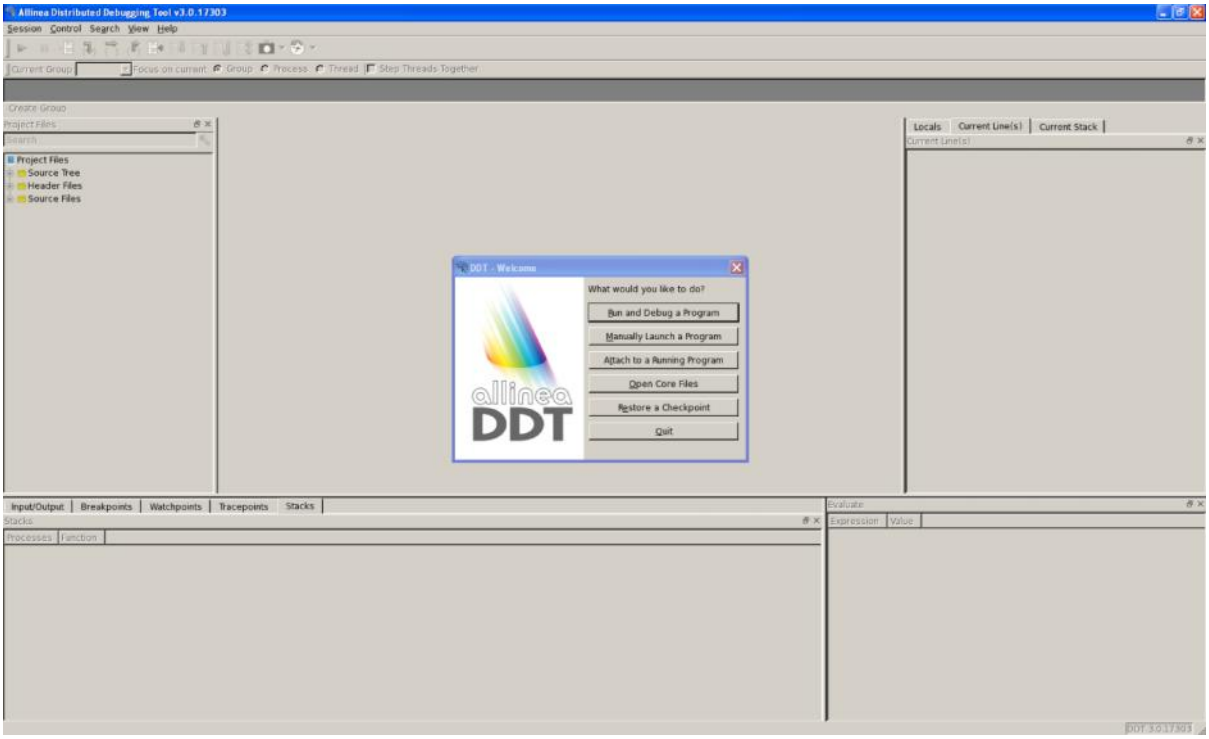


Figure 4: DDT Welcome Screen.

7.2.5 Submitting a job through DDT

- To submit a job to the LSF queue for debugging select the Run and Debug a Program button from the Welcome Screen.
- In the DDT-Run (queue Submission mode) window (Figure 5).
    - The program to run is specified in the Application box. It is possible to select the code to run by clicking the folder icon and browsing to the location of the executable.
    - Pass any arguments to the program via the Arguments box.
    - Next select the MPI library to use by choosing the correct LSF template script. This should match the MPI library used when compiled the program. By default the template file for the Intel MPI library will be used. If you wish to use Platform MPI library template script, click the Change button on the Options: line. In the Job Submission Settings window, change the Submission template file by clicking the folder icon beside this box and selecting the lsf_platformMPI.qtf file (Figure 6).

- Note if the user requires any pre-processing before or post-processing after debugging the program, they should copy the appropriate LSF template file from /app/tools/allinea/ddt/templates to their home directory and edit it as appropriate. Then use this file as the Submission template file.

- Now set the number of MPI processes to use. This is controlled by two parameters - the number of processes per node and the number of nodes. The default value for the number of processes per node is 12 (i.e. nodes are fully occupied). To change this click the Change… button on the Options: line. In the window under Job Submission Settings change the value of PROC_PER_NODE_TAG to change the number of MPI processes per node (see Figure 7). Click OK and set the number of nodes required as in Figure 8 (the total number of MPI processes is then number of nodes * number of processes per node).

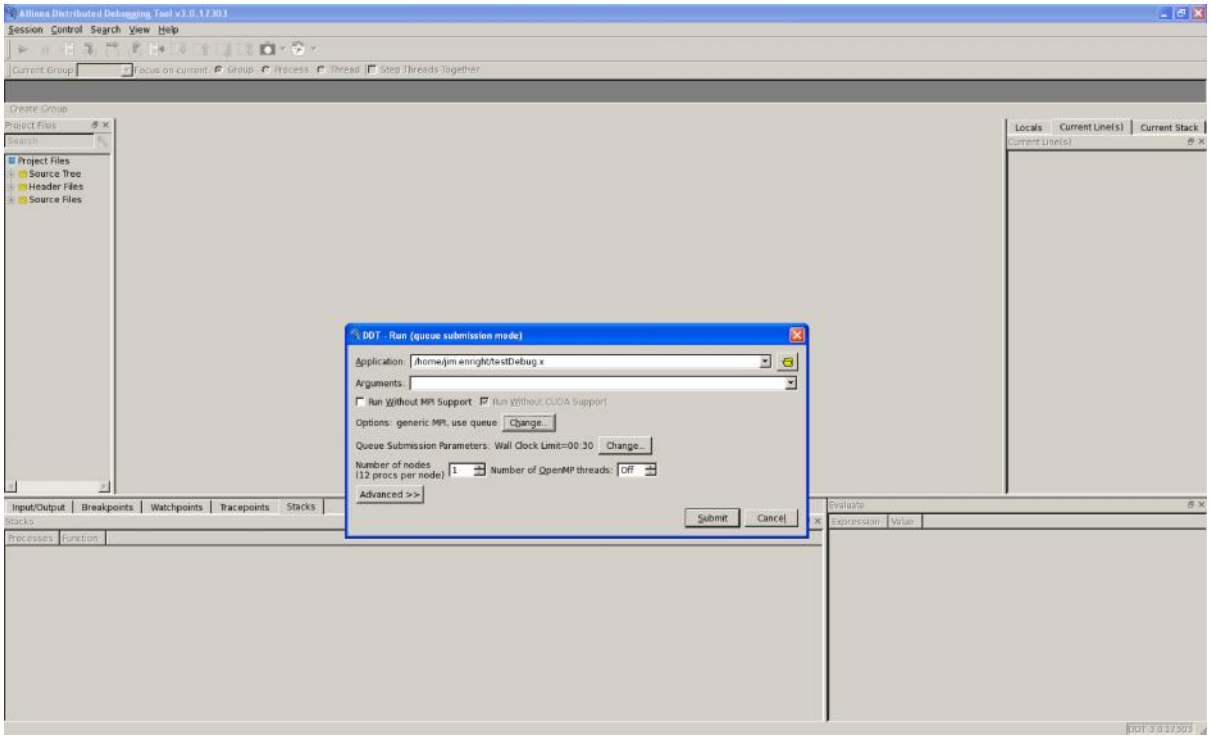


Figure 5: The DDT-Run (queue Submission mode) window. The program to run is put in the Application box; any arguments to the program are put in the Arguments box.

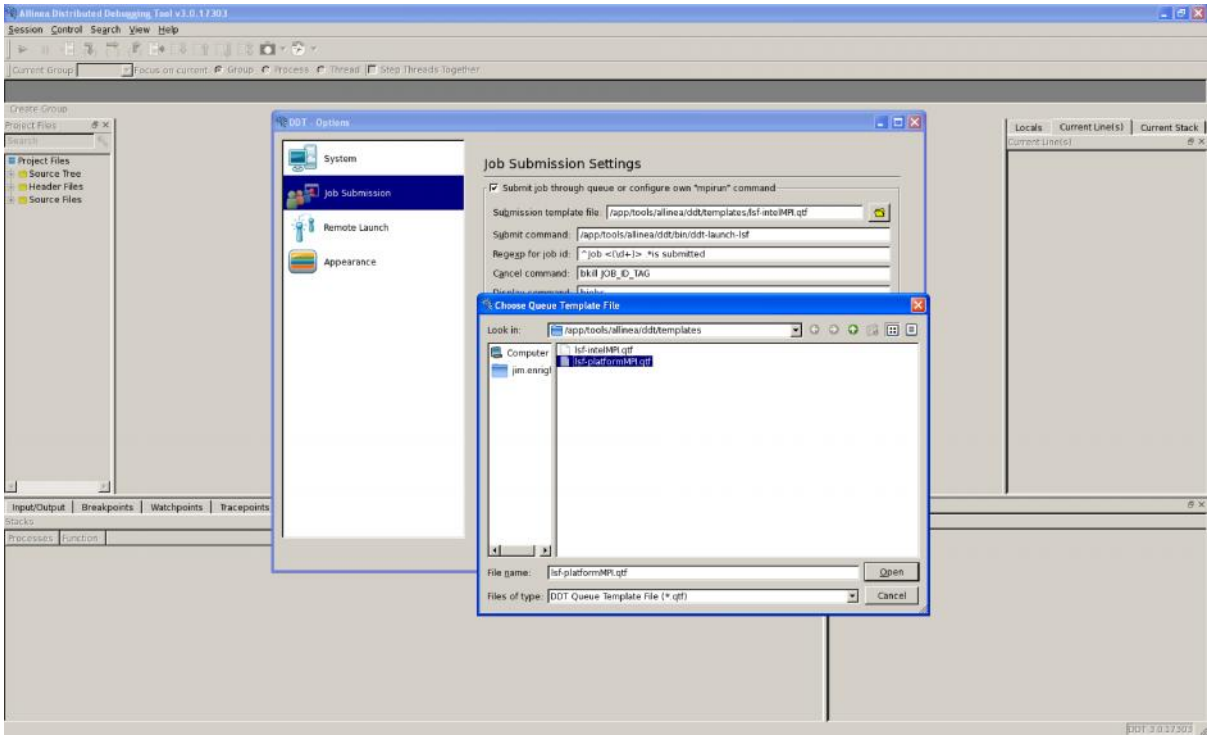


Figure 6: Changing the MPI library by selecting the correct LSF template script.

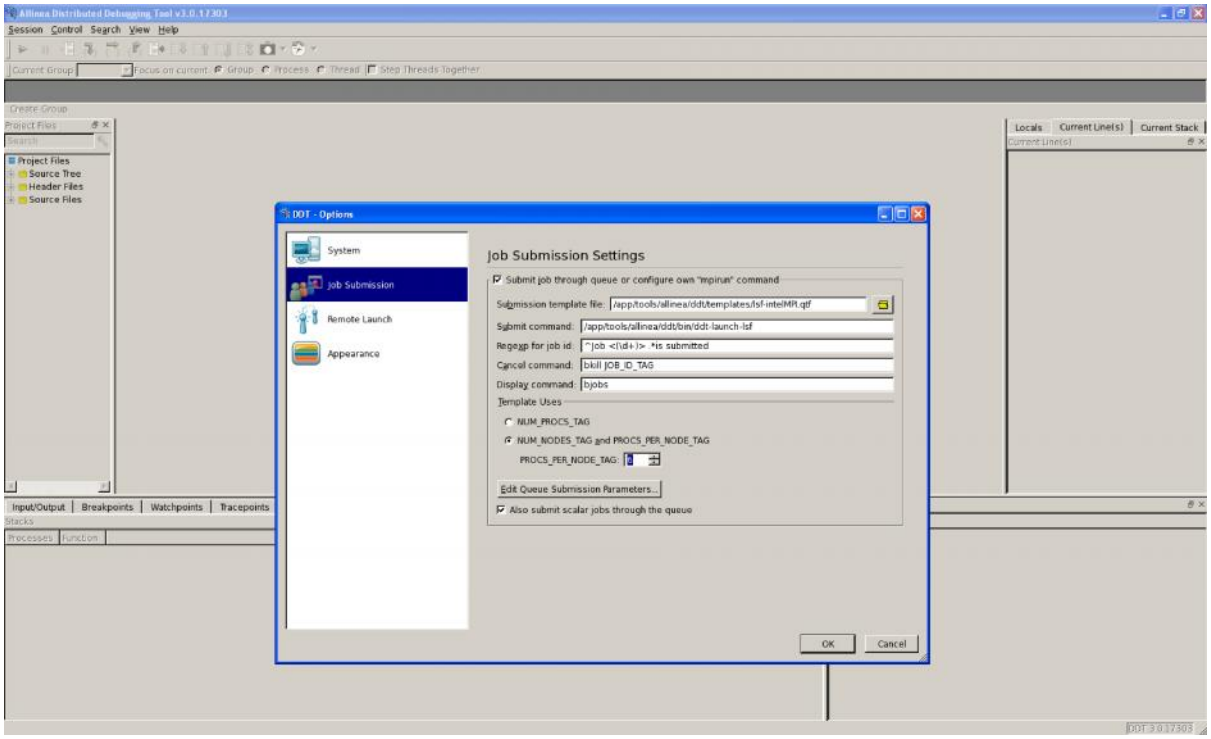


Figure 7: Changing the number of MPI processes per node by changing the PROC_PER_NODE_TAG value. Here we use 6 MPI processes per node.

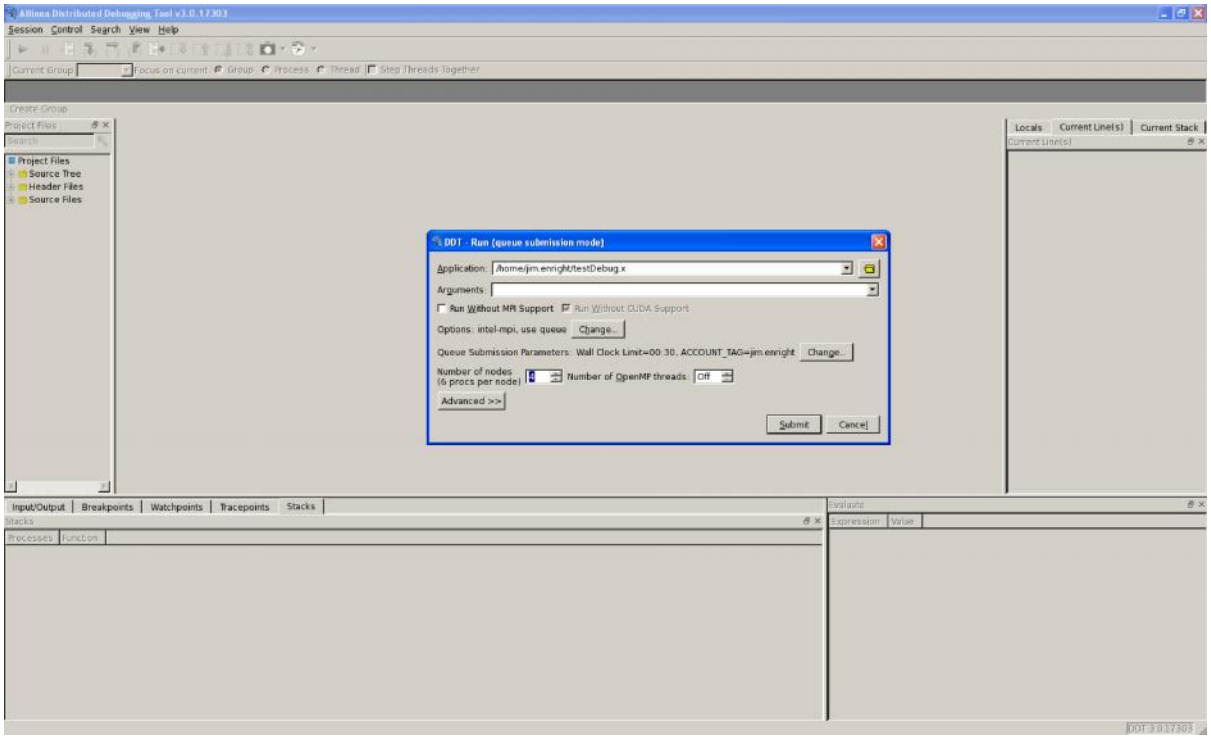


Figure 8: Set the number of nodes to use - here we use 4 nodes. With 6 processes per node, this gives a total of 4*6 = 24 MPI processes.

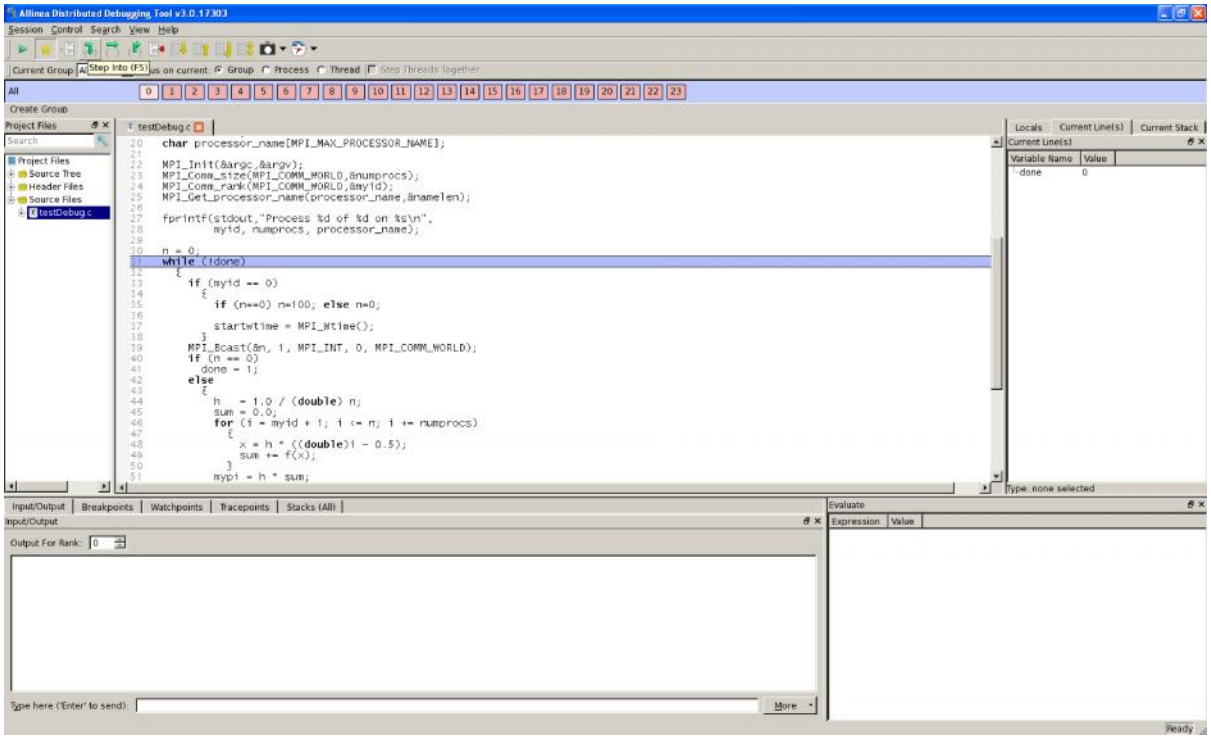


Figure 9: DDT attaches to the running processes when they are launched. The program can then be debugged in the DDT GUI.

- Click Submit to put the job in the queue.
- Allinea DDT now waits until the job is run from the queue. DDT will then attach to the running processes and debugging can start.

7.2.6 Debugging the program using DDT

See the DDT User Guide, available at

/app/doc/tools/allinea/ddt/usermanual.pdf

for instructions on debugging the program once submitted.

8 Job Control

As each cluster has a finite set of resources, users' jobs are managed by a scheduler which holds the jobs in a queue until the job's requested resources become available.

A **resource** in HPC terms is a unit of operation, be it CPU cores, memory, or length of time the job should run.

Once you have successfully built the program which will be executed on the cluster, you will need to create a simple script which contains the parameters for the scheduler. This **job script** specifies what computing resources your job needs, and provides the scheduler the instructions required to execute your job.

The scheduler is derived from Platform's Load Sharing Facility (or **LSF**).

8.1 Job submission

8.1.1 Run Script

In all cases, you will need to create a **run script** which specifies everything that the scheduler needs in order to execute your job.

As each HPC Wales sub-system has slightly different capabilities and job control system e.g. queue names, the details of job submission varies a little. In its simplest form, job submission may be accomplished completely via a simple single line command that can be submitted to run on the compute nodes (see 8.1.2). In practice we recommend supplying all details of the run within the run script itself. This point will become clearer as we progress.

The run script is a standard Unix shell script containing special directives resembling shell comments which contain instructions for the scheduler. Below is an example with some common options which are explained beneath.

The examples here use **bash** for the run script, but you can use your preferred shell.

```
#!/bin/bash --login

#BSUB -q queue_name
#BSUB -J job_name
#BSUB -W walltime=hh:mm:ss
#BSUB -n cores

Program Arguments
```

As shown above, users of the bash shell are advised to use `#!/bin/bash --login` at the start of their run script as the `--login` flag ensures that the run-time environment will be set up as for a normal login.

queue_name

The name of the queue in which your job should be placed. The available queues differ depending on which system you are using, and are explained later.

job_name

You can assign a short meaningful name to your job, which will be displayed to any user who checks the job queue. If you omit this option, the executable name will be used instead.

walltime

Walltime is the amount of time that your job will need to complete. You should try to estimate the shortest length of time that your job should run, in order not to hog the system for too long. If your job has not stopped by the time the walltime period has expired, it will be killed. Being more accurate with your choice of walltime can also help the scheduler run your job earlier.

n

The number of cores your job requests.

program arguments

program is the name of the executable that launches your job; **arguments** are any parameters that the command requires. Information about how to start your program is shown in the examples below.

The main job queue on the Cardiff HTC cluster is called "q_cf_htc_work". Shown below is a run script similar to the one below, substituting the resource limits for those that a typical job requires.

```
#!/bin/bash --login

#BSUB -q q_cf_htc_work
#BSUB -J Test
#BSUB -W walltime=1:00:00
#BSUB -n 24

Program Arguments
```

8.1.2 Submitting the Job

To submit the job to the queuing system, use the `bsub` command:

```
$ bsub < your_run_script
```

Progress of your job can then be checked as described in “Job Monitoring and Control”

Some common parameters, including those mentioned above, which can be passed to the **bsub** command are summarised in Table 6 below.

NOTE: Users familiar with other job schedulers e.g., PBSPro may be used to submitting jobs under control of the qsub directive through a simple command such as

```
$ qsub your_run_script
```

Note at this point that this syntax should NOT be used on the HPC Wales sub systems. Although the scheduler will accept such a syntax, and appear to have successfully submitted the job, *processing of the* job will ignore all of the #BSUB directives contained within the script, leading to unpredictable results. *This is illustrated briefly below:*

```
$ bsub < Gromacs_d.dppc-cutoff_submit.ppn=12.MFG.q
Job <551129> is submitted to default queue <q_cf_htc_work>.
```

```
$ qsub  Gromacs_d.dppc-cutoff_submit.ppn=12.MFG.q
Job <551131> is submitted to default queue <q_cf_htc_work>.
Request <551131> is submitted to default queue <q_cf_htc_work>.
```

While the above response to the `bsub` and `qsub` commands appear similar, note the additional line associated with the incorrect use of qsub. Note also that the job submitted under control of qsub will ignore the BSUB request for 128 cores.

```
#!/bin/bash --login
#BSUB -n 128
#BSUB -x
#BSUB -W 1:00
#BSUB -o results/LOGS.MFG/Gromacs.ppn=12.o.%J
#BSUB -e results/LOGS.MFG/Gromacs.ppn=12.e.%J
#BSUB -J Gromacs
#BSUB -R "span[ptile=12]"
```

and attempt to run the job on a single processor!

Parameter	Description
`-n min_proc[,max_proc]`	Specify number of CPU cores required
`-R “span[ptile=N]”`	Specify number of processes per node
`-W [hour:]minute`	Time limit for job
`-x`	Exclusive access to compute nodes
`-J jobname`	Specify a job name
`-o outputfile`	Specify location for standard output
`-e errorfile`	Specify location for standard error

Table 6. Common parameters that can be passed to the bsub command

8.1.3 Resource Limits

Note that there is a walltime limit of 72 hours for jobs. This is to ensure a decent turnover of jobs.

8.2 Program Execution

The important part of your job submission script is the instruction to the scheduler for executing your program. In most cases this will be the final line of your script.

Almost certainly the code execution instruction will need to know how many processes to start on the cluster; this will typically be a product of the nodes and processors per node, but

it's trivial to have the submission script calculate this value for you. Various environment variables are set by LSF during a job and you can use these variable names in defining just how your program is to be executed. These include those listed in Table 7 below.

Environment Variable	Description
`LSB_JOBID`	Job ID
`LSB_DJOB_NUMPROC`	Number of processes
`LSB_DJOB_HOSTFILE`	Location of hostfile
`LSB_MCPU_HOSTS`	Hostname information in format: host1 n1 host2 n2 host3 n3 ...
`LSB_HOSTS`	Hostname information in format: host1 host1 host1 host2 host2 host2 ...
`LSB_JOBNAME`	Job name

Table 7. Various environment variables set by LSF during a job

The job scheduler will create a list of nodes on which the job will execute, and compute the number of processes requested – given by the environment variable pointed to by the environment variable LSB_DJOB_NUMPROC.

For a program written using the MPI libraries, the program should be executed with the **mpirun** command, which takes as its arguments the number of processes to run and the program code and its arguments. Note that it is in general advisable to specify the full path to your executable.

```
mpirun -np $LSB_DJOB_NUMPROC progname args
```

... where *progname* is the name of your executable, and *args* are the arguments that your program requires.

Note that by default the job's current working directory will be that which that contains your job script; this can of course be changed by using the `cd` command within the job script itself prior to the execution of the mpirun command e.g.

```
WDPATH=/scratch/$USER/$LSB_JOBID
rm -rf $WDPATH
mkdir -p $WDPATH
cd ${WDPATH}
```

Beware that if you have compiled your code with the Intel compilers, it will be necessary to ensure the appropriate environment module is also loaded by your submission script.

Commercial software packages are often launched in ways which are specific to each package. Some even require arguments which relate to the structure of the submitted model data.

We have some examples for popular commercial software on the Example Run Scripts page but in general you will need to carefully check the manual for your particular software package.

8.3 Example Run Scripts

8.3.1 Example 1

A simple single line command can be submitted to run on the compute nodes using the following syntax:

```
$ bsub –n 1 –o output.%J command [options]
```

where –n specifies the number of CPU cores required and the –o flag specifies the output file. Note that when specifying the output file with –o outout.%J, then %J is replaced with the Job ID number.

8.3.2 Example 2

The following more complicated example runs an executable compiled with the Intel compiler and Intel MPI:

```
#!/bin/bash --login
#BSUB –o example.o.%J
#BSUB –x
#BSUB –n 24             # Number of cores to use
#BSUB –W 1:00           # 1 hour time limit
#BUSB –q q_cf_htc_work  # Submit to a specific queue
# load the intel compiler and MPI modules
module purge
module load compiler/intel
module load mpi/intel
MYPATH=$HOME/mycode_directory
executable=$MYPATH/bin/my_executable_name
# Run in specified directory WDPATH
WDPATH=$MYPATH/RESULTS
cd ${WDPATH}
# $LSB_DJOB_NUMPROC is supplied by LSF and is number of processes
mpirun –np $LSB_DJOB_NUMPROC $executable
```

Note that the appropriate modules needed by the intel-compiled code are specified first, while the executable is assumed to reside in a different directory to the directory in which the job will be run.

8.3.3 Example 3

The following builds on Example 2, but now runs the job in the Lustre global scratch directory belonging to the user (/scratch/$USER), deleting that directory on completion of the job:

```
#!/bin/bash --login
#BSUB –o example.o.%J
#BSUB –x
#BSUB –n 24             # Number of cores to use
#BSUB –W 1:00           # 1 hour time limit
#BUSB –q q_cf_htc_work  # Submit to a specific queue
```

```
module purge
module load compiler/intel
module load mpi/intel
MYPATH=$HOME/mycode_directory
executable=$MYPATH/bin/my_executable_name

# Run in Lustre global scratch directory
WDPATH=/scratch/$USER/$LSB_JOBID
rm -rf $WDPATH
mkdir -p $WDPATH
cd ${WDPATH} || exit $?
trap "rm -rf ${WDPATH}" EXIT # delete scratch directory on exit

# $LSB_DJOB_NUMPROC is supplied by LSF and is number of processes

mpirun –np $LSB_DJOB_NUMPROC $executable
```

Note the use of `$LSB_JOBID` to create a unique lustre sub-directory in which to run the job

8.3.4 Example 4

The following example mirrors Example 3, running the job in the Lustre global scratch directory belonging to the user (`/scratch/`$USER), but now illustrates the use of `pdsh` to a run a command on all the compute nodes, here running the `memhog` utility to clear memory of the nodes to provide optimal and consistent performance:

```
#!/bin/bash --login
#BSUB –o example.o.%J
#BSUB –x
#BSUB –n 24             # Number of cores to use
#BSUB –W 1:00           # 1 hour time limit
#BUSB –q q_cf_htc_work  # Submit to a specific queue

module purge
module load compiler/intel
module load mpi/intel
MYPATH=$HOME/mycode_directory
executable=$MYPATH/bin/my_executable_name

# Run in Lustre global scratch directory
WDPATH=/scratch/$USER/$LSB_JOBID
rm -rf $WDPATH
mkdir -p $WDPATH
cd ${WDPATH} || exit $?
trap "rm -rf ${WDPATH}" EXIT # delete scratch directory on exit

# generate list of hosts for use by pdsh
PDSH_HOSTS=`echo  $LSB_MCPU_HOSTS  |  awk  '{for(i=1;i<NF;i=i+2)  printf
$i",";}'`

# pdsh can be used to run a command on all compute nodes of a job
# for example when benchmarking one can run memhog to
# clear memory to provide optimal and consistent performance

pdsh -w $PDSH_HOSTS memhog 35g > /dev/null 2>&1
```

```
# $LSB_DJOB_NUMPROC is supplied by LSF and is number of processes

mpirun –np $LSB_DJOB_NUMPROC $executable
```

8.3.5 Example 5. Execution of the DLPOLY classic Code

The following example illustrates execution of the DLPOLY classic molecular simulation code under control of the associated module `dlpoly-classic/1.8`. The job is run using the Lustre global scratch directory belonging to the user (`/scratch/$USER`), with the DLPOLY input data sets residing in the user's directory `DLPOLY-classic/data/Bench5` initially copied into the scratch directory `/scratch/$USER/DLPOLY-classic.$LSB_JOBID`: note the use of the LSF parameter `$LSB_JOBID` to create a unique descriptor for that directory.

```
#!/bin/bash --login

#BSUB -n 128
#BSUB -x
#BSUB -o Bench5.HTC.o.%J
#BSUB -e Bench5.HTC.e.%J
#BSUB -J Bench5
#BSUB -R "span[ptile=12]"
#BSUB -W 1:00
#BSUB -q q_cf_htc_work

module load compiler/intel-11.1.072
module load mpi/intel-4.0
module load dlpoly-classic/1.8

export OMP_NUM_THREADS=1

code=$DLPOLY_EXECUTABLE
MYPATH=${HOME}/DLPOLY-classic/data
MYDATA=${HOME}/DLPOLY-classic/data/Bench5
WDPATH=/scratch/$USER/DLPOLY-classic.$LSB_JOBID

NCPUS=$LSB_DJOB_NUMPROC

env
rm -rf ${WDPATH}
mkdir ${WDPATH}
cd ${WDPATH}
# copy input files to working directory
cp -r -p ${MYDATA}/CONFIG  ${WDPATH}
cp -r -p ${MYDATA}/CONTROL ${WDPATH}
cp -r -p ${MYDATA}/FIELD   ${WDPATH}
cp -r -p ${MYDATA}/TABLE   ${WDPATH}
rm -f REVCON  REVIVE  STATIS

echo "CPUS=$NCPUS, NODES=$NNODES"

time mpirun -r ssh -np $NCPUS ${code}
# copy output file back to home filestore
cp ${WDPATH}/OUTPUT  ${MYPATH}/Bench5.out.impi
cd ${WDPATH}
```

```
rm -f REVCON  REVIVE  STATIS OUTPUT
```

The simulation output is subsequently copied back into the user's directory as file `Bench5.out.impi` on completion of the job. The user is referred to section 9 of this manual to compare the use of SynfiniWay when running the DLPOLY classic.

8.3.6 Compiling and running OpenMP threaded applications

Code with embedded OpenMP directives may be compiled and run on a single compute node with up to a maximum of NCORES threads via the smp parallel environment, where NCORES is the number of CPU cores per node.

Compiling OpenMP code

OpenMP code may be compiled with the Intel compiler and with `gcc/gfortran version >= 4.2`. Here is a simple example from the tutorial at http: //openmp.org/wp/

```
C******************************************************************************
C FILE: omp_hello.f
C DESCRIPTION:
C   OpenMP Example - Hello World - Fortran Version
C   In this simple example, the master thread forks a parallel region.
C   All threads in the team obtain their unique thread number & print it.
C   The master thread only prints the total number of threads.  Two OpenMP
C   library routines are used to obtain the number of threads and each
C   thread's number.
C AUTHOR: Blaise Barney
C LAST REVISED:
C******************************************************************************

      PROGRAM HELLO
      INTEGER NTHREADS, TID, OMP_GET_NUM_THREADS,
     +        OMP_GET_THREAD_NUM
C     Fork a team of threads giving them their own copies of variables
!$OMP PARALLEL PRIVATE(NTHREADS, TID)
C     Obtain thread number
      TID = OMP_GET_THREAD_NUM()
      PRINT *, 'Hello World from thread = ', TID
C     Only master thread does this
      IF (TID .EQ. 0) THEN
        NTHREADS = OMP_GET_NUM_THREADS()
        PRINT *, 'Number of threads = ', NTHREADS
      END IF
C     All threads join master thread and disband
!$OMP END PARALLEL
      END
```

Here are basic compile options for Gnu and Intel compilers for the Fortran code (C code same - but substitute corresponding C compiler in each case).

Gnu: `gfortran -fopenmp -o hello omphello.f`

Intel: `ifort -openmp -o hello omphello.f`

Running OpenMP code

To run an OpenMP code, a job script needs to be created, and submitted it to the smp parallel environment. Using the hello example above here is a script called run.sh

```
#!/bin/bash --login
#BSUB -n 4
#BSUB -x
#BSUB -o OpenMP.HTC.rx600.o.%J
#BSUB -J HELLO
#BSUB -R "span[ptile=4]"
#BSUB -W 1:00
#BSUB -q q_cf_htc_large
# latest intel compilers
module load compiler/intel-11.1.072

export OMP_NUM_THREADS=4
code=${HOME}/linux_openmp/source_code_2/hello
MYPATH=$HOME/linux_openmp/source_code_2

TESTS="OpenMP"

cd ${MYPATH}
echo running OpenMP TEST OMP_NUM_THREADS=$LSB_DJOB_NUMPROC
export OMP_NUM_THREADS=$LSB_DJOB_NUMPROC

${code}
```

To submit the script:

```
$ bsub < run.sh
Job <202999> is submitted to queue < q_cf_htc_large>.
```

The output file run.o.202999 after the job has completed:

```
running OpenMP TEST OMP_NUM_THREADS=4
 Hello World from thread =            0
 Number of threads =            4
 Hello World from thread =            2
 Hello World from thread =            1
 Hello World from thread =            3
```

There are two points to note. Firstly the second line of the run.sh script contains q_cf_htc_large. This ensures that the script is submitted to the SMP parallel environment. Secondly the environment variable OMP_NUM_THREADS should be set to the number of required threads. It is recommended the value does not exceed the number of cores/CPUs per node. If the OMP_NUM_THREADS variable is not set then the default value depends in principle on which compiler was used – in practice both Gnu and Intel compilers set this to the number of cores.

8.4 Job Monitoring and Control

There are a variety of LSF commands for monitoring the status of jobs, and for assessing the progress of a given job and overall job usage. These include the `bjobs`, `bpeek`, `bkill`, `bqueues` and `bacct` commands, each of which is summarised below.

8.4.1 The `bjobs` command

The command `bjobs` shows the status of your jobs in the queue:

```
$ bjobs
JOBID    USER      STAT   QUEUE         FROM_HOST    EXEC_HOST    JOB_NAME
SUBMIT_TIME
202885   martyn.  RUN    q_cf_htc_b  log001    12*htc028    Bench4       Feb   5
16:26
                                              4*htc122
202886   martyn.  RUN    q_cf_htc_b  log001   12*htc096    Bench7       Feb   5
16:26
                                              4*htc049
202887   martyn.  RUN    q_cf_htc_b  log001   12*htc081    CPMD         Feb   5
16:29
                                             12*htc082
                                             12*htc083
                                             12*htc110
```

Adding the option "`-u all`" shows the status of the jobs of all users:

```
$ bjobs -u all
JOBID    USER      STAT   QUEUE         FROM_HOST    EXEC_HOST    JOB_NAME
SUBMIT_TIME
202850   andy.th  RUN    q_cf_htc_b  log002       12*htc070    GAMESS       Feb   5
13:50
                                             12*htc071
                                             12*htc072
                                             12*htc100
202851   andy.th  RUN    q_cf_htc_b  log002       12*htc073    GAMESS       Feb   5
13:51
                                             12*htc101
                                             12*htc074
                                             12*htc102
202865   christo  RUN    q_cf_htc_b  log001       12*htc054    *th_3c_101 Feb   5
14:39
                                             12*htc056
                                             12*htc057
                                             12*htc058
                                             12*htc059
                                             12*htc001
```

```
                                                12*htc002
                                                12*htc004
202867  christo RUN   q_cf_htc_b log001      12*htc013   TIP4P_200  Feb  5
15:42
202868  christo RUN   q_cf_htc_b log001      6*htc021    TIP4P_225  Feb  5
15:43
202869  christo RUN   q_cf_htc_b log001      6*htc014    TIP4P_250  Feb  5
15:44
202870  christo RUN   q_cf_htc_b log001      6*htc051    TIP4P_275  Feb  5
15:46
202883  christo RUN   q_cf_htc_b log001      12*htc109   *P4P_375_L Feb  5
16:01
                                                12*htc091
202884  christo RUN   q_cf_htc_b log001      12*htc048   *P4P_400_L Feb  5
16:02
                                                12*htc080
202885  martyn. RUN   q_cf_htc_b log001      12*htc028   Bench4     Feb  5
16:26
                                                4*htc122
202886  martyn. RUN   q_cf_htc_b log001      12*htc096   Bench7     Feb  5
16:26
                                                4*htc049
202887  martyn. RUN   q_cf_htc_b log001      12*htc081   CPMD       Feb  5
16:29
                                                12*htc082
                                                12*htc083
                                                12*htc110
```

To find more detail about a specific job then use the **-l** flag with required Job ID:

```
$ bjobs -l 600319

Job <600319>, Job Name <muc100>, User <georgina.menzies>, Project <default>, St
                     atus <RUN>, Queue <q_cf_htc_work>, Job Priority <50>, Comm
                     and <#!/bin/bash --login ;#BSUB -x;#BSUB -n 60;#BSUB -o mu
                     c100;#BSUB -e muc100;#BSUB -J muc100;#BSUB -W 72:00;#BSUB
                     -R "span[ptile=12]";  NPROC=36;  # Load the Environment; m
                     odule purge ;module load use.own;module load Gromacs_4.5.5
                     -single_GM;   # Run the Program;grompp_mpi -f em.mdp -c io
                     n.gro -p topol.top -o em.tpr -maxwarn 5;mpirun -np ${LSB_D
                     JOB_NUMPROC:-1} mdrun_mpi -s em.tpr -c em.gro -o em.trr -e
                      em.edr -g em.log;grompp_mpi -f pr.mdp -c em.gro -p topol.
                     top -o pr.tpr -maxwarn 5;mpirun -np ${LSB_DJOB_NUMPROC:-1}
                      mdrun_mpi -s pr.tpr -c pr.gro -o pr.trr -e pr.edr -g pr.l
                     og;grompp_mpi -f md.mdp -c pr.gro -p topol.top -o md.tpr -
                     maxwarn 5;mpirun -np ${LSB_DJOB_NUMPROC:-1} mdrun_mpi -s m
                     d.tpr -c md.gro -o md.trr -e md.edr -g md.log>, Share grou
                     p charged </georgina.menzies>
Mon Mar 25 14:33:42: Submitted from host <cf-log-001>, CWD <$HOME/mucin/Muc100-
                     nogly>, Output File <muc100>, Exclusive Execution, Re-runn
                     able, 60 Processors Requested, Requested Resources <span[p
                     tile=12]>;

 RUNLIMIT
 4320.0 min of cf-htc-076
Tue Mar 26 08:03:24: Started on 60 Hosts/Processors <12*cf-htc-076> <12*cf-htc-
                     107> <12*cf-htc-014> <12*cf-htc-137> <12*cf-htc-091>, Exec
                     ution Home </home/georgina.menzies>, Execution CWD </home/
```

```
                     georgina.menzies/mucin/Muc100-nogly>;
Fri Mar 29 03:24:04: Resource usage collected.
                     The CPU time used is 11151953 seconds.
                     MEM: 4952 Mbytes;  SWAP: 16266 Mbytes;  NTHREAD: 256
                     PGID: 10781;  PIDs: 10781 10782 10786 6082 6083 6084 6085
                     6086 6087 6088 6089 5996 6090 6091 6092 6093 6094 6095
                     6096 6101 6097 6098 6099
                     PGID: 9684;  PIDs: 9684 9685 9686 9687 9688 9689 9690
                     9691 9692 9693 9694 9695 9696
                     PGID: 7561;  PIDs: 7561 7562 7563 7564 7565 7566 7567
                     7568 7569 7570 7571 7572 7573
                     PGID: 12148;  PIDs: 12148 12149 12150 12151 12152 12153
                     12154 12155 12156 12157 12158 12159 12160
                     PGID: 5498;  PIDs: 5498 5499 5500 5501 5502 5503 5504
                     5505 5506 5507 5508 5509 5510


 SCHEDULING PARAMETERS:
           r15s   r1m  r15m   ut      pg    io   ls    it    tmp    swp    mem
 loadSched   -     -     -     -       -     -    -     -     -      -      -
 loadStop    -     -     -     -       -     -    -     -     -      -      -
```

8.4.2 The bpeek command

The **bpeek** command can be used to show the output from a particular job. If no Job ID is specified then the latest job is shown, in the case shown below from running a GAMESS electronic structure job:

```
<< output from stdout >>
running NCPUs= PPN=12
----- GAMESS execution script 'rungms' -----
This job is running on host htc086.htc.hpcwales.local
under operating system Linux at Sun Feb 5 16:43:19 GMT 2012
Available scratch disk space (Kbyte units) at beginning of the job is
Filesystem           1K-blocks      Used Available Use% Mounted on
192.168.128.217@o2ib:192.168.128.218@o2ib:/scratch
                     183138519936 8161617256 173141327292   5% /scratch
Copying input file bench1.inp to your run's scratch directory...
MPI kickoff will run GAMESS on 24 cores in 24 nodes.
The binary to be executed is /home/mike/gamess/gamess.00.x
MPI will run 24 compute processes and 24 data servers,
    placing 12 of each process type onto each node.
The scratch disk space on each node is /scratch/username/GAMESS.202889, with
free space
Filesystem           1K-blocks      Used Available Use% Mounted on
192.168.128.217@o2ib:192.168.128.218@o2ib:/scratch
                     183138519936 8161617256 173141327292   5% /scratch
          ************************************************
          *         GAMESS VERSION = 11 AUG 2011 (R1)          *
          *             FROM IOWA STATE UNIVERSITY             *
          * M.W.SCHMIDT, K.K.BALDRIDGE, J.A.BOATZ, S.T.ELBERT, *
          *   M.S.GORDON, J.H.JENSEN, S.KOSEKI, N.MATSUNAGA,   *
          *           K.A.NGUYEN, S.J.SU, T.L.WINDUS,          *
          *       TOGETHER WITH M.DUPUIS, J.A.MONTGOMERY       *
          *         J.COMPUT.CHEM.  14, 1347-1363(1993)        *
          ************** 64 BIT INTEL VERSION **************
```

```
<< output from stderr >>
```

8.4.3 The bkill command

If you wish to cancel a job which has been submitted then use the **bkill** command with the appropriate Job ID:

```
$ bkill 202887
Job <202887> is being terminated
```

If you wish to cancel all your jobs in the queue then use the command "**bkill 0**"

8.4.4 The bqueues command

The status of the available queues can be shown with the **bqueues** command:

```
$ bqueues
QUEUE_NAME      PRIO STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN  SUSP
dynamic_provisi  60  Open:Active       -    -    -    -     6     6     0     0
q_cf_htc_work    30  Open:Active       -  768    -    -  6012  4494  1518     0
q_cf_htc_1024    30  Open:Active       -    -    -    -  1024  1020     0     0
q_cf_htc_intera  30  Open:Active       -    -    -    -     0     0     0     0
q_cf_htc_large   25  Open:Active       -    -    -    -     0     0     0     0
q_cf_htc_vlarge  25  Open:Active       -    -    -    -     0     0     0     0
q_cf_htc_win     25  Open:Active       -    -    -    -     0     0     0     0
```

8.4.5 The bacct command

The `bacct` command displays a summary of accounting statistics for all finished jobs (with a DONE or EXIT status) submitted by the user who invoked the command, on all hosts, projects, and queues in the LSF system. `bacct` displays statistics for all jobs logged in the current LSF accounting log file:

```
$ bacct
Accounting information about jobs that are:
  - submitted by users username,
  - accounted on all projects.
  - completed normally or exited
  - executed on all hosts.
  - submitted to all queues.
  - accounted on all service classes.
------------------------------------------------------------------------------
SUMMARY:      ( time unit: second )
 Total number of done jobs:    4206      Total number of exited jobs:   196
 Total CPU time consumed:   129419.2      Average CPU time consumed:    29.4
 Maximum CPU time of a job: 30115.4       Minimum CPU time of a job:     0.0
 Total wait time in queues: 33922324.0
 Average wait time in queue: 7706.1
 Maximum wait time in queue:126331.0      Minimum wait time in queue:    2.0
 Average turnaround time:      8397 (seconds/job)
 Maximum turnaround time:    129932       Minimum turnaround time:         2
 Average hog factor of a job:  0.04 ( cpu time / turnaround time )
```

```
Maximum hog factor of a job:  15.09      Minimum hog factor of a job:  0.00
Total throughput:             1.38 (jobs/hour)  during 3198.12 hours
Beginning time:       Sep 25 11:40      Ending time:          Feb  5 16:47
```

8.5 Interactive Jobs

It is sometimes desirable from a system management point of view to control all workload through a single centralized scheduler. Thus running an interactive job through the LSF batch system allows you to take advantage of batch scheduling policies and host selection features for resource-intensive jobs. You can submit a job and the least loaded host is selected to run the job.

Since all interactive batch jobs are subject to LSF policies, you will have more control over your system. For example, you may dedicate two servers as interactive servers, and disable interactive access to all other servers by defining an interactive queue that only uses the two interactive servers.

8.5.1 Scheduling policies

Running an interactive batch job allows you to take advantage of batch scheduling policies and host selection features for resource-intensive jobs. Note that an interactive batch job is scheduled using the same policy as all other jobs in a queue. This means an interactive job can wait for a long time before it gets dispatched. If fast response time is required, interactive jobs should be submitted to high-priority queues with loose scheduling constraints.

8.5.2 Submitting Interactive Jobs

Finding out which queues accept interactive jobs

Before you submit an interactive job, you need to find out which queues accept interactive jobs with the bqueues -l command.

If the output of this command contains the following, this is a batch-only queue. This queue does not accept interactive jobs:

```
SCHEDULING POLICIES:  NO_INTERACTIVE
```

If the output contains the following, this is an interactive-only queue:

```
SCHEDULING POLICIES:  ONLY_INTERACTIVE
```

If none of the above are defined or if SCHEDULING POLICIES is not in the output of bqueues -l, both interactive and batch jobs are accepted by the queue.

As can be seen from the response below, the queue `q_cf_htc_interactive` on the Cardiff HTC system is targeted for interactive work.

```
QUEUE: q_cf_htc_interactive
  -- Interactive queue to run jobs on BX900 blades
PARAMETERS/STATISTICS
PRIO NICE STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN SSUSP USUSP  RSV
```

```
 30    0  Open:Active       -    -    -    -     0     0     0     0     0    0
SCHEDULING PARAMETERS
           r15s   r1m  r15m   ut      pg    io   ls    it    tmp    swp    mem
 loadSched   -     -     -     -       -     -    -     -     -      -      -
 loadStop    -     -     -     -       -     -    -     -     -      -      -
SCHEDULING POLICIES:  BACKFILL  EXCLUSIVE  ONLY_INTERACTIVE
USERS: all
HOSTS:  cf_htc_bx900/
RERUNNABLE :  yes
```

Submit an interactive job by using a pseudo-terminal

8.5.3 bsub -Is

1. To submit a batch interactive job and create a pseudo-terminal with shell mode support, use the `bsub -Is` option. For example:

```
$ bsub -Is csh
Queue does not accept interactive jobs. Job not submitted.
```

A reminder – note that the default queue does not support interactive work, so as noted above, please use q_cf_htc_interactive

```
$ bsub -Is -q q_cf_htc_interactive csh
Job <203673> is submitted to queue <q_cf_htc_interactive>.
<<Waiting for dispatch ...>>
<<Starting on htc057>>
[username@htc057 examples]$ exit
exit
```

This will submit a batch interactive job that starts up csh as an interactive shell.

When you specify the `-Is` option, `bsub` submits a batch interactive job and creates a pseudo-terminal with shell mode support when the job starts. This option should be specified for submitting interactive shells, or applications which redefine the CTRL-C and CTRL-Z keys (for example, jove).

2. The following example shows how to run the DLPOLY_4 application in interactive mode. First, we show below the corresponding LSF script, then present the sequence of interactive steps that replicate the batch job.

```
#!/bin/bash --login
#BSUB -n 32
#BSUB -x
#BSUB -o DLPOLY4.test2.HTC.o.%J
#BSUB -J DLPOLY4
#BSUB -R "span[ptile=8]"
#BSUB -W 1:00
```

```
#BSUB -q q_cf_htc_work

module purge
module load compiler/intel-11.1.072
module load mpi/intel-4.0

export OMP_NUM_THREADS=1

code=${HOME}/training_workshop/DLPOLY4/execute/DLPOLY.Z
MYPATH=$HOME/training_workshop/DLPOLY4/examples
WDPATH=$MYPATH/TMP

NCPUS=$LSB_DJOB_NUMPROC
_tmp=($LSB_MCPU_HOSTS)
PPN="${_tmp[1]}"
REPEAT="0"
TEST="TEST2"
rm -r -f ${WDPATH}
mkdir ${WDPATH}

cp -r -p ${MYPATH}/$TEST/*  ${WDPATH}/
cd ${WDPATH}
for i in $REPEAT; do
    echo running TEST=$TEST NCPUs=$NCPUS PPN=$PPN REPEAT=$i
    time mpirun -r ssh -np $NCPUS ${code}
    mv ${WDPATH}/OUTPUT
${MYPATH}/test2.out.HTC.impi.n${NCPUS}.PPN=$PPN.${i}
      rm -f REVCON  REVIVE  STATIS
done
```

Now the interactive jobs, following the steps given below:

1. initiate the interactive session, requesting 32 cores – 4 nodes (using ptile=8) - and wait for the session to commence

```
$ pwd
/home/username/training_workshop/DLPOLY4/examples
$ bsub -Is -q q_cf_htc_interactive -n 32 -R "span[ptile=8]" bash
Job <203671> is submitted to queue <q_cf_htc_interactive>.
<<Waiting for dispatch ...>>
<<Starting on htc081>>
```

2. Now change to the working directory, copy across the input files needed, and trigger the mpirun command, requesting 32 cores

```
$ echo $LSB_JOBID
$ rm -rf /scratch/$USER/DLPOLY4.$LSB_JOBID
$ mkdir /scratch/$USER/DLPOLY4.$LSB_JOBID
$ cp -rp TEST2/* /scratch/$USER/DLPOLY4.$LSB_JOBID
$ cd /scratch/$USER/DLPOLY4.$LSB_JOBID
$ mpirun -np 32 $HOME/training_workshop/DLPOLY4/execute/DLPOLY.Z
```

3. now copy the output file to its final destination, and exit from the interactive shell

```
[username@htc081 DLPOLY4.203671]$ ls -ltr
total 133364
-rw------- 1 username username      457 May  3  2002 FIELD
-rw------- 1 username username 63072365 Oct  5  2010 CONFIG
-rw------- 1 username username      454 Dec 24 17:29 CONTROL
-rw-rw-r-- 1 username username     3562 Feb 12 23:11 STATIS
-rw-rw-r-- 1 username username 63072365 Feb 12 23:11 REVCON
-rw-rw-r-- 1 username username 13864344 Feb 12 23:11 REVIVE
-rw-rw-r-- 1 username username    26371 Feb 12 23:11 OUTPUT
$ cp -rip OUTPUT
$HOME/training_workshop/DLPOLY4/examples/test2.out.HTC.impi.n32.PPN=8
$ exit
exit
```

8.5.4 Submit an interactive job and redirect streams to files

bsub -i, -o, -e

You can use the -I option together with the -i, -o, and -e options of bsub to selectively redirect streams to files. For more details, see the bsub(1) man page.

1. To save the standard error stream in the job.err file, while standard input and standard output come from the terminal:
2. `$ sub -I -q q_cf_htc_interactive -e job.err lsmake`

Split stdout and stderr

If in your environment there is a wrapper around bsub and LSF commands so that end-users are unaware of LSF and LSF-specific options, you can redirect standard output and standard error of batch interactive jobs to a file with the > operator.

By default, both standard error messages and output messages for batch interactive jobs are written to stdout on the submission host.

1. To write both stderr and stdout to mystdout:

```
bsub -I myjob 2>mystderr 1>mystdout
```

2. To redirect both stdout and stderr to different files, set LSF_INTERACTIVE_STDERR=y in lsf.conf or as an environment variable.

   For example, with LSF_INTERACTIVE_STDERR set:

   bsub -I myjob 2>mystderr 1>mystdout

   stderr is redirected to mystderr, and stdout to mystdout.

See the *Platform LSF Configuration Reference* for more details on LSF_INTERACTIVE_STDERR.

9 Using the SynfiniWay Framework

9.1 Access methods

There are three ways for end-users to work with the SynfiniWay framework:

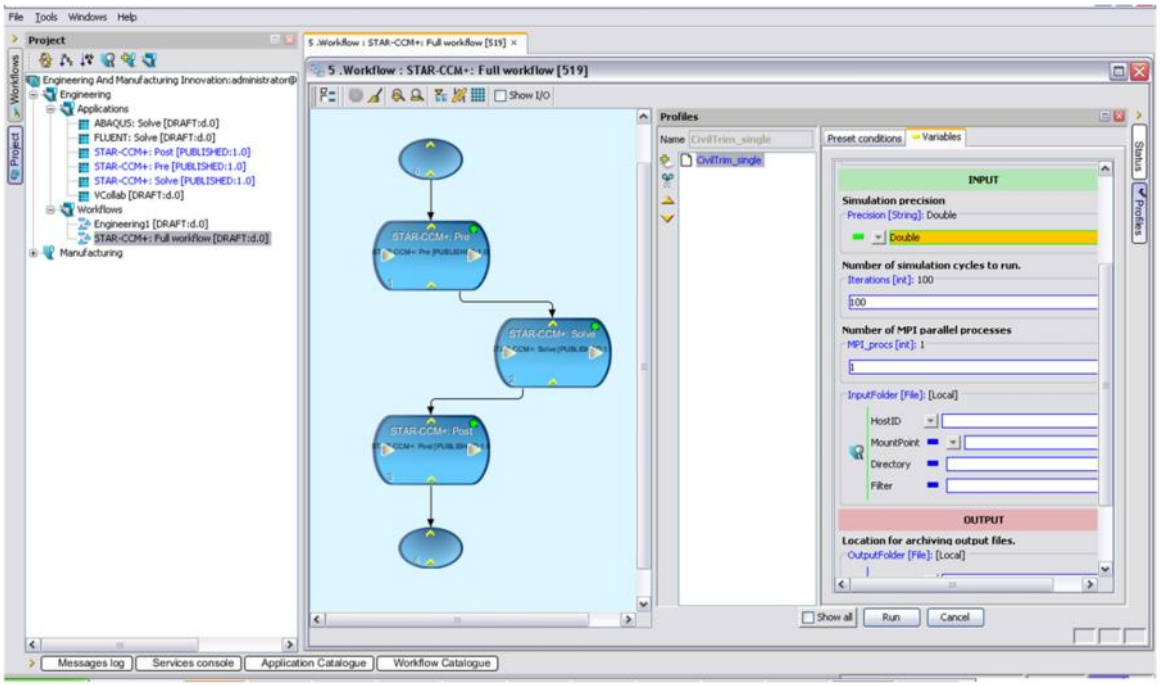
- **Web interface**
- **Java client interface**
- **Command line utilities**

9.1.1 SynfiniWay web interface

The web interface is reached by navigating through the HPC Wales Portal to one of the Gateway pages. Each gateway has a link to the SynfiniWay web portal assigned to that gateway. SynfiniWay will provide access only to workflows provided for the selected gateway.

SynfiniWay manages its own authentication mechanism. When connecting through the HPC Wales Portal this authentication is implicit, and a single login challenge is issued only at the initial portal connection for each session. The account credentials for using the portal are the same as for a direct SSH session on any HPC Wales system. The other two access methods will require explicit authentication.

9.1.2 SynfiniWay Java Client



The SynfiniWay Java Client is a graphical utility whose primary purpose is workflow development and management. It can be easily downloaded and installed on a desktop system by individual users. Use of this utility may be limited by to individuals with workflow editing rights as defined by the HPC Wales global security policy, and implemented through the SynfiniWay RBAC mechanism. A full description of the Java Client will eventually be provided in a separate section below.

The SynfiniWay Java client provides access to the complete development functionality for encapsulating HPC business processes as automated SynfiniWay workflows. In addition it includes a *project* area for organizing workflows, applications and profiles of the SynfiniWay objects catalogue in a tree-structured manner. The notion of such projects is arbitrary, meaning they can map to other organisational entities, such as disciplines themes, domains, or actual broader definitions of project. This capability is utilised to organise applications and workflow that are presented and accessed through the research gateways in the portal. A sample view of the Java client is shown in the above image.

More details of the SynfiniWay Java Client are given in section 9.6.

9.1.3 SynfiniWay user line commands

Command-line utilities will generally be accessible to any end-user. These will be provided on each of the HPC Wales sub-system login nodes.

The set of SynfiniWay line commands are shown in the following table.

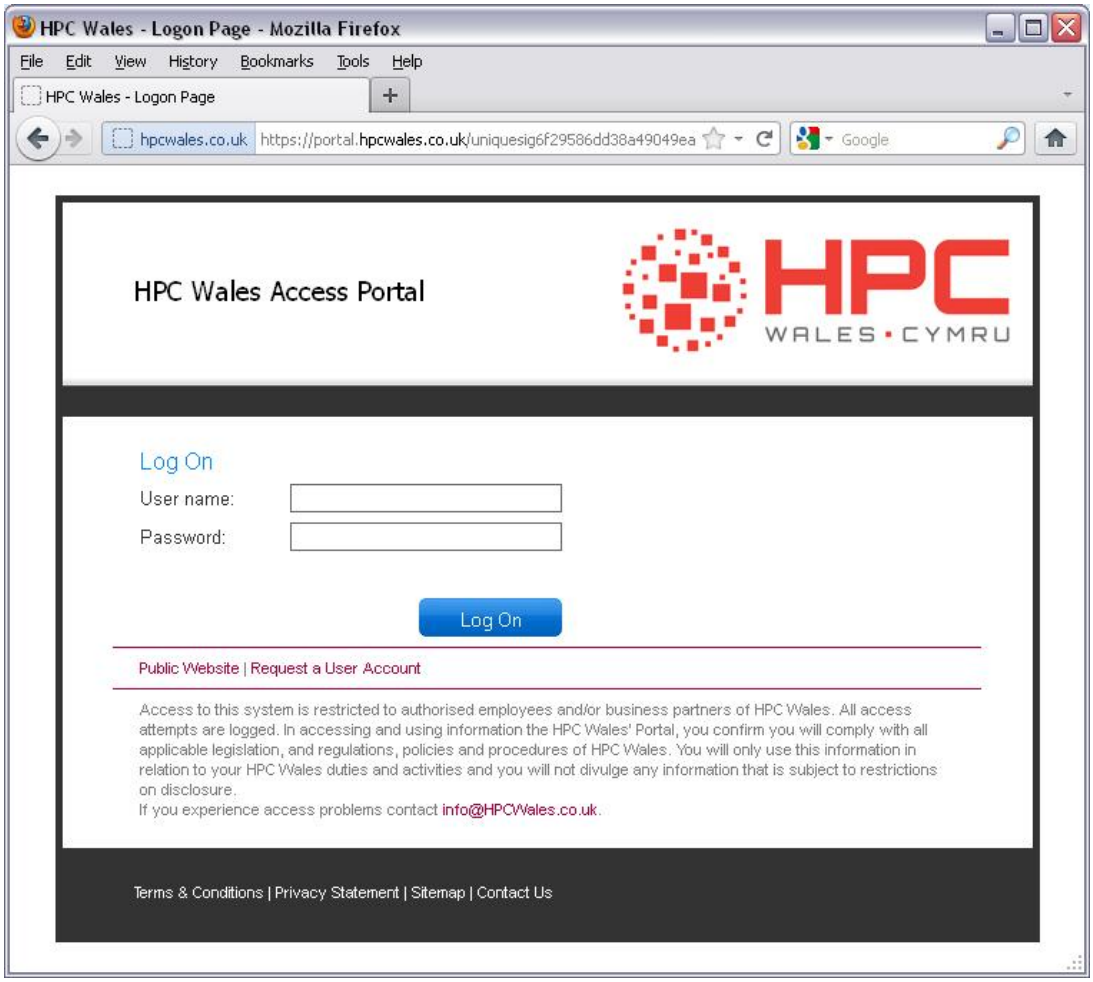
Command	Description
sfy_cat	Dump the file content in the current standard output
sfy_ls	Listing
sfy_cp	File copy
sfy_mkdir	Create directory
sfy_rm	Delete file
sfy_del	Delete workflow
sfy_kill	Kill workflow
sfy_sub	Submit workflow
sfy_stat	Workflow(s) status

Additional command may be added in future releases of the SynfiniWay software.

9.2 HPC Wales Portal

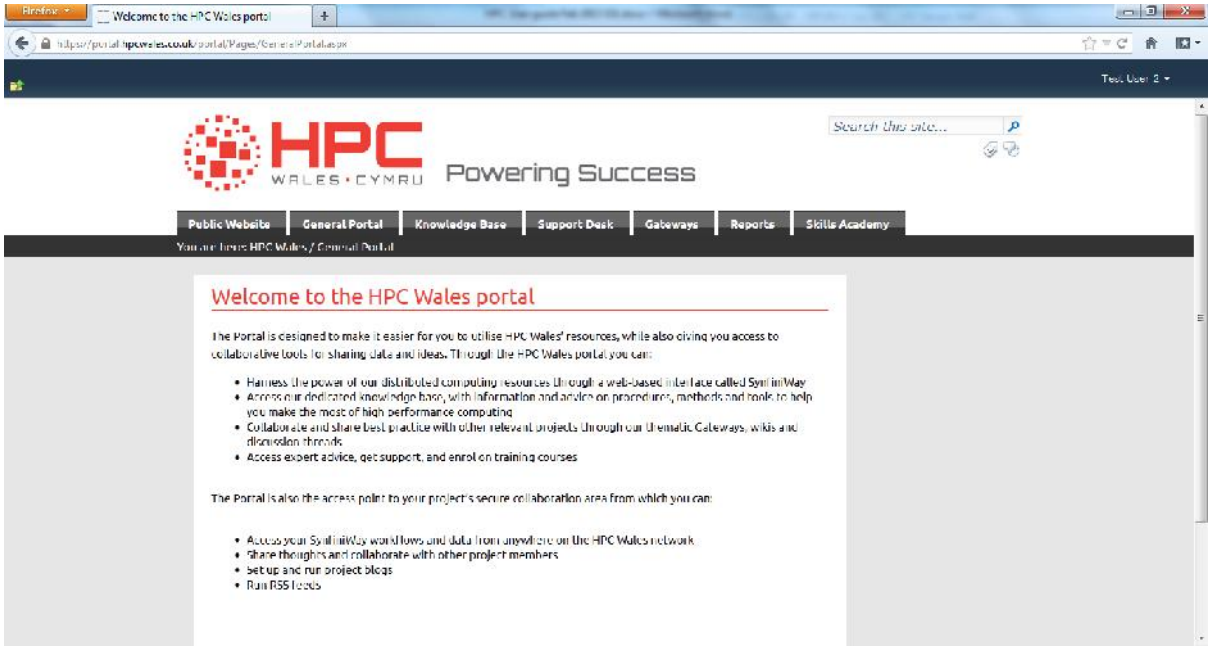
9.2.1 Entering the HPC Wales Portal

The HPC Wales Portal can be located at the URL: https://portal.hpcwales.co.uk


HPC Wales - Logon Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
HPC Wales - Logon Page
hpcwales.co.uk https://portal.hpcwales.co.uk/uniquesig6f29586dd38a49049ea
Google
HPC Wales Access Portal
HPC WALES · CYMRU
Log On
User name:
Password:
Log On
Public Website | Request a User Account
Access to this system is restricted to authorised employees and/or business partners of HPC Wales. All access attempts are logged. In accessing and using information the HPC Wales' Portal, you confirm you will comply with all applicable legislation, and regulations, policies and procedures of HPC Wales. You will only use this information in relation to your HPC Wales duties and activities and you will not divulge any information that is subject to restrictions on disclosure.
If you experience access problems contact info@HPCWales.co.uk.
Terms & Conditions | Privacy Statement | Sitemap | Contact Us

When accessing this URL you will be challenged for your HPC Wales login account. At the panel above enter your normal account credentials.

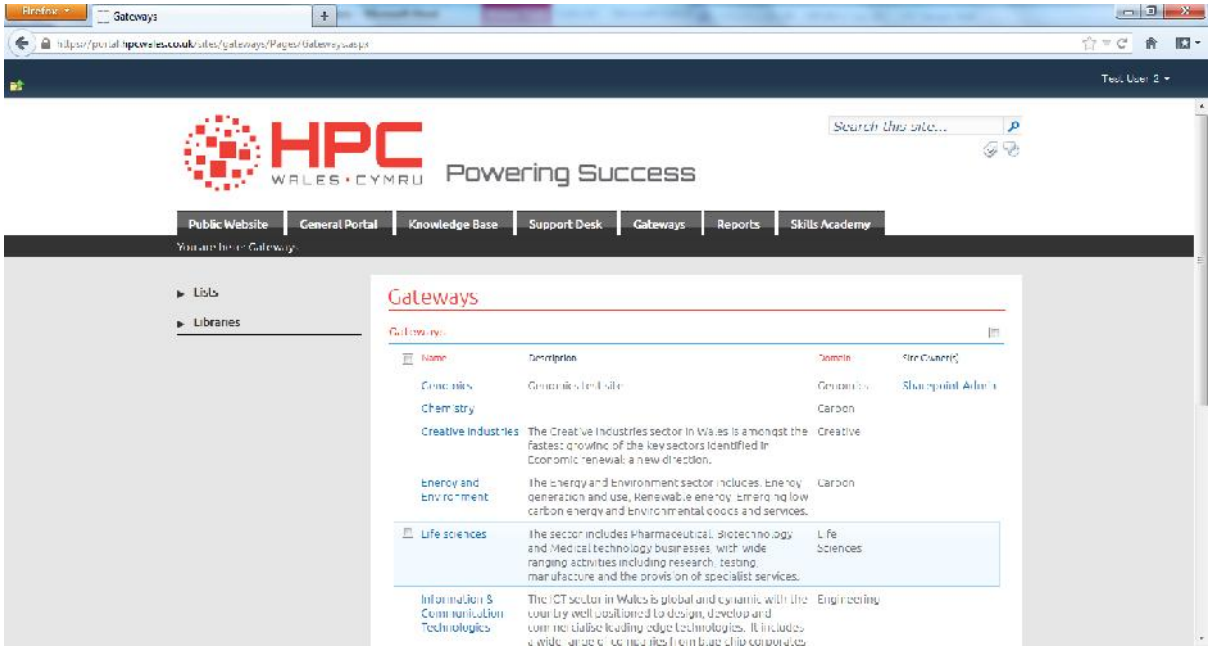
If the authentication succeeds then you will observe the main entry page for HPC Wales.



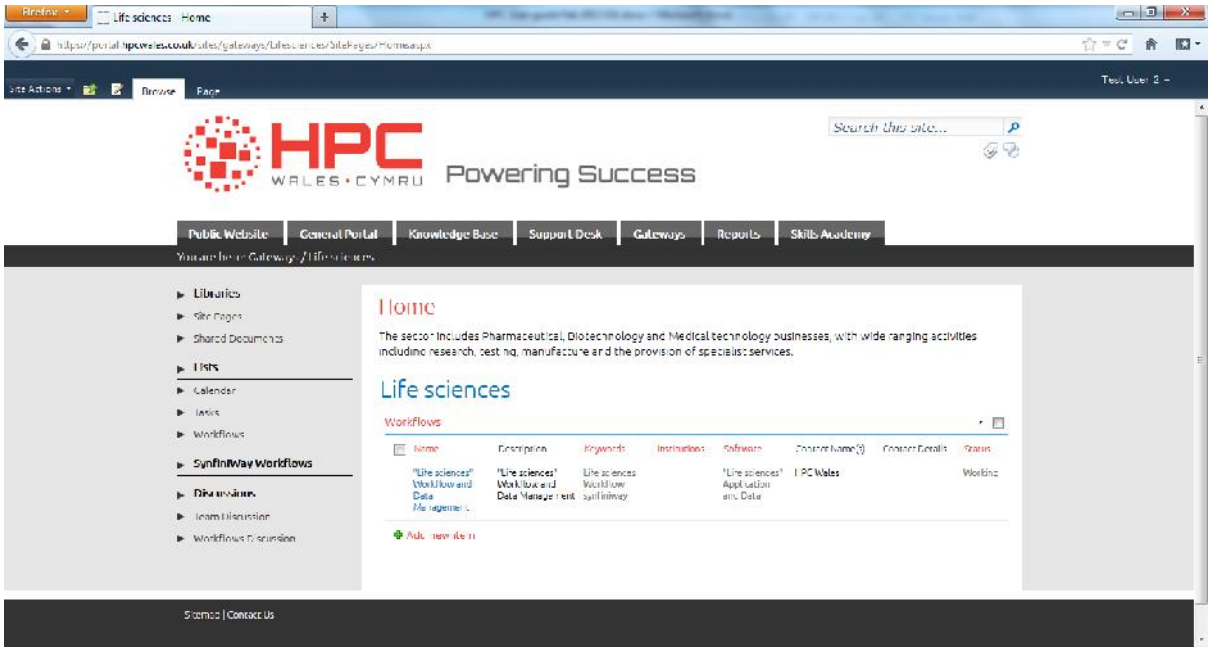
By selecting the **Gateways** tab you will be presented with the list of domain-specific gateways available at HPC Wales. Currently entry points for five gateways are defined. The definition and number of gateways will evolve as more disciplines and projects are included in the HPC Wales service.

9.2.2 Opening a Gateway

To open a gateway click on the relevant name in the **Gateways** page. The current list of gateways is shown in the next image.



In the following sections the Life Sciences gateway will be used to illustrate use of HPC Wales systems through the web interface. Opening the **Life Sciences** gateway will display the following launch page.

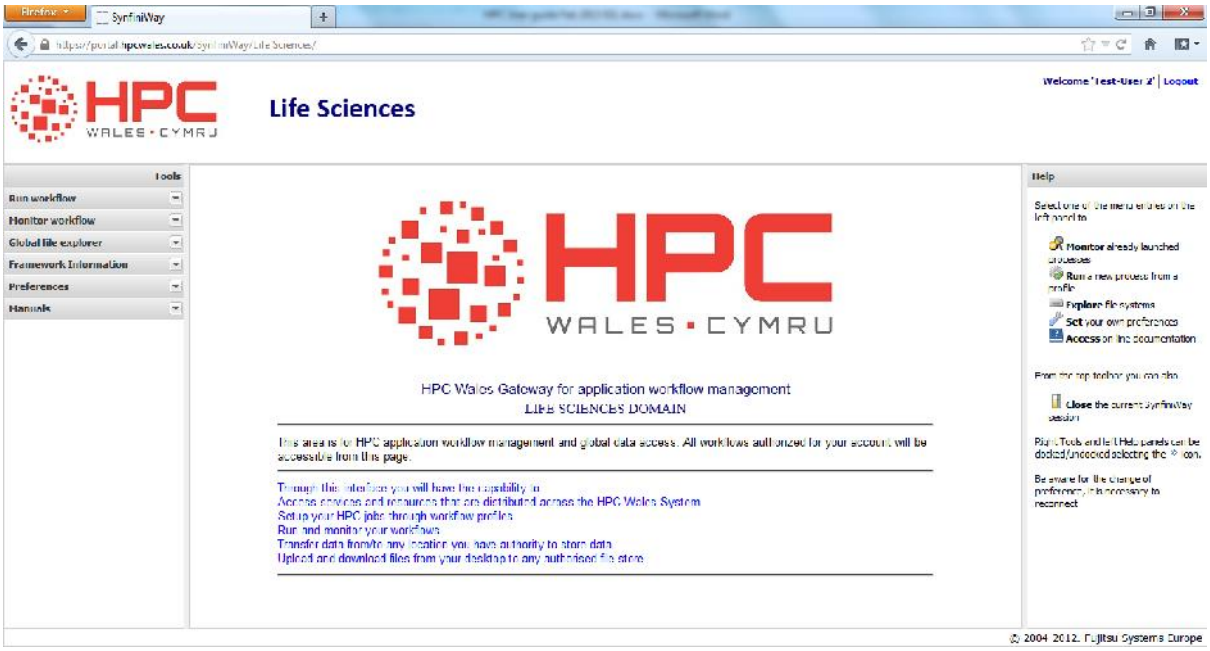


The content of the gateway home page is hosted within the HPC Wales Microsoft Sharepoint system. Use of the various functions on such pages is described in other documents. This guide is concerned with running applications and manipulating input and output files across the HPC Wales global system, as provided by the SynfiniWay framework.

Each gateway has a separate SynfiniWay web portal. Connecting to this portal is done by clicking on the either of these links in the thematic or project gateway home page:

- **SynfiniWay Workflows**
- **<Theme or project name> Workflow and Data Management**

Following these links from the Life Sciences home page will open the following SynfiniWay workflow portal. Similar portals are accessible from the other gateway home pages.

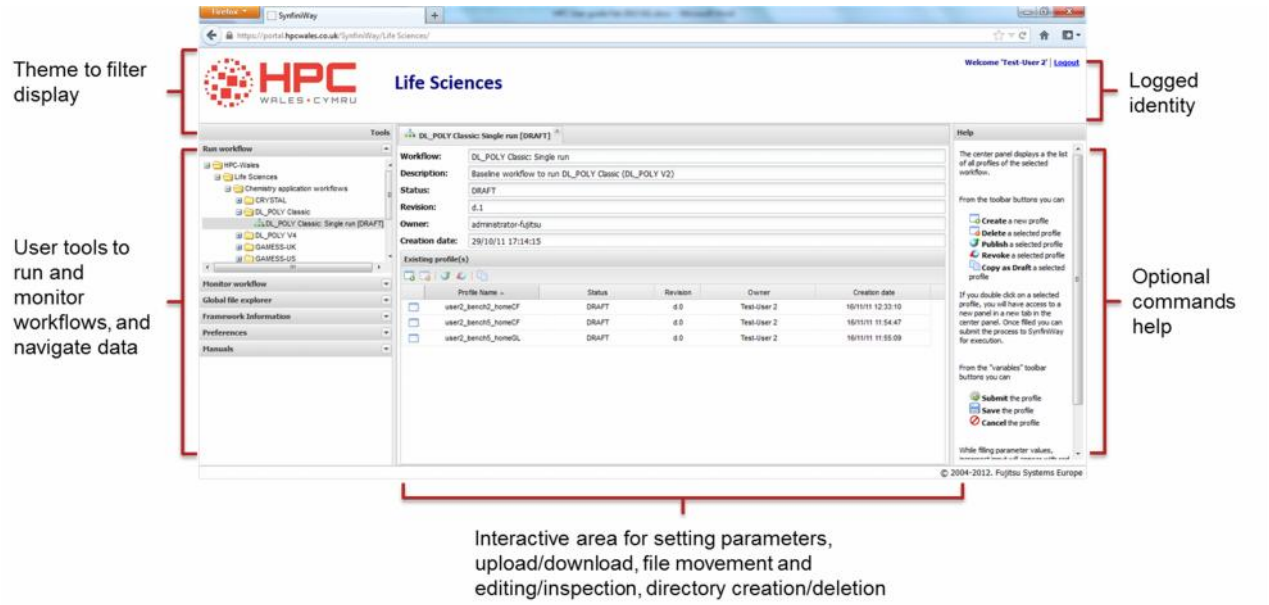


This view is the starting point for using the HPC Wales system to run packaged application workflows, as described in the following sections.

9.3 SynfiniWay Gateway

9.3.1 Page layout

The SynfiniWay page has five panels.


Theme to filter display
Logged identity
User tools to run and monitor workflows, and navigate data
Optional commands help
Interactive area for setting parameters, upload/download, file movement and editing/inspection, directory creation/deletion

Users will work only within the left and centre panels. The help panel can be hidden by setting the configuration in the **Preferences** tool.

Panel	Purpose
Top	Shows Gateway name and user identity.
Left	Toolset to run and monitor workflow, view and handle data, and obtain general framework information.
Centre	Workflow parameter entry forms, workflow monitoring output and data explorer view.
Right	Contextual help.
Bottom	Copyright statement.

The **Tools** area is the starting point for end-user work. The different tools provided are opened in turn through an accordion display. The following sub-sections introduce each of the tools currently installed.

9.3.2 Tools: Run workflow

Workflows within the selected gateway theme will be displayed in this area.

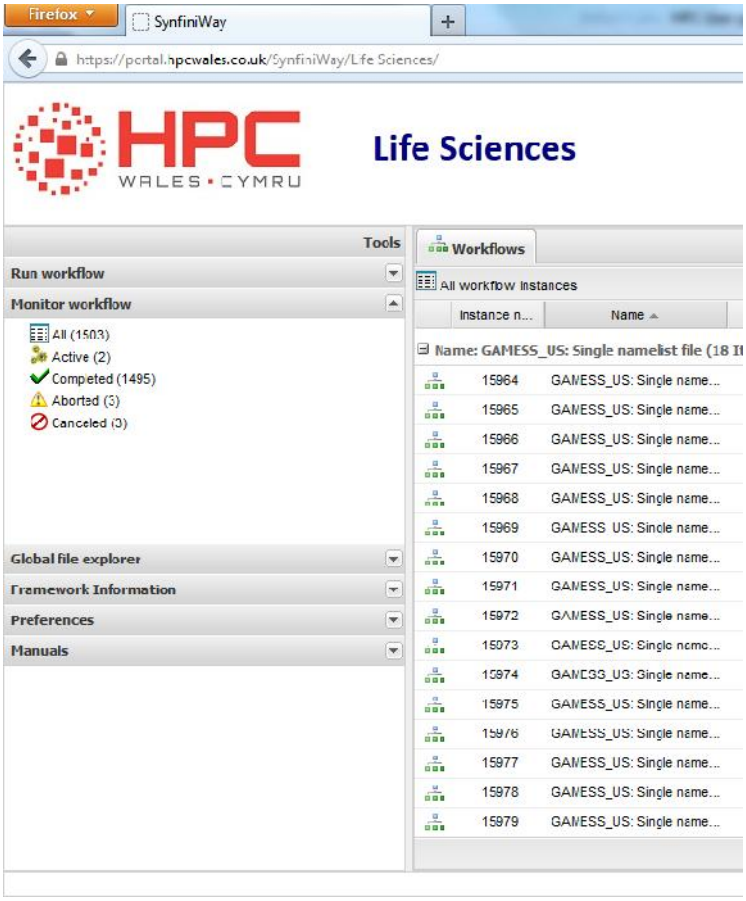
Workflows are organised within a hierarchical folder structure. A single first-level folder is common to all gateways, and fixed to the name **HPC-Wales**. By convention the second-level

folder name is the same as the gateway. Third-level folders and below are used only to facilitate workflow organisation, and there is no restriction on their name or structure.

The above image shows workflows grouped around a subset of chemistry applications installed on the HPC Wales sub-systems: In each case multiple workflows are currently made available. Each workflow encodes a distinct process that can be run according to the end-user needs.

9.3.3 Tools: Monitor workflow

Monitoring allows you to check the progress of submitted workflows and the functional status of workflow tasks. The first level entry point to the monitor area shows the global view of workflow activity.



To avoid saturation of the overall system a clean-up procedure is advised for terminated workflows, whether normally or abnormally. The monitor area only shows workflows that have not been cleaned. Workflow clean-up is described in a separate section below.

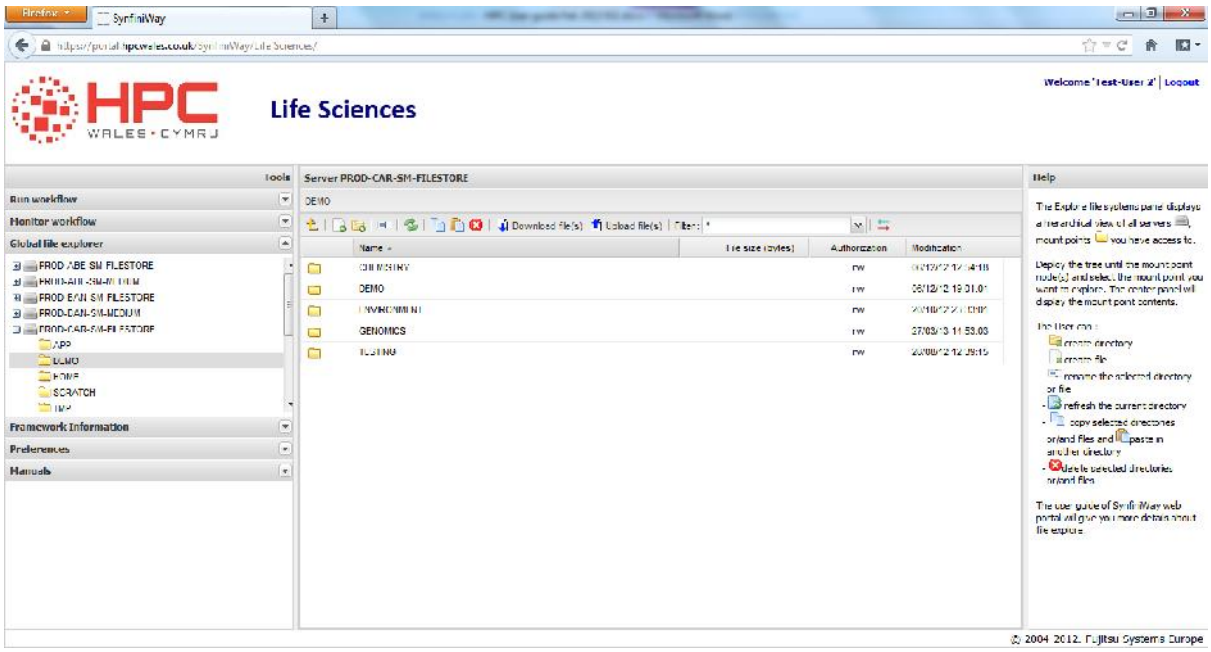
9.3.4 Tools: Global file explorer

The Global File Explorer provides a single point of access to the file systems on any HPC Wales site that are mounted on a SynfiniWay SM. This access has two parts:

a) List of SynfiniWay mount point in the Tools window.

b) List of files and directories under the selected mount point in the centre panel of the web view.

An image from within the Life Sciences gateway is shown below.



SynfiniWay mount points map to a directory or folder within a file-system. Such mount points can map to any level in the file-system hierarchy. Below the mount point user can navigate to any location where permitted by the OS native access control. Navigation above the mount point is not possible.

The initial set of mount points is described in the next table. Each HPC Wales sub-system will have the same mount points.

SynfiniWay mount point name	Description
SHARED	Maps to the /shared directory area on each site.
APP	Maps to the /app directory area on each site.
TMP	Maps to the /tmp directory area on each machine hosting the designated SM.
HOME	Maps to the /home area for the authenticated login of this portal session.
SCRATCH	Maps to the /scratch area on each site or sub-system.

Permission to view and use SynfiniWay mount point is regulated by the RBAC mechanism. The initial RBAC setup allows all users to see the complete set of general baseline mount points. The only mount point that is currently explicitly controlled is HOME. Other mount

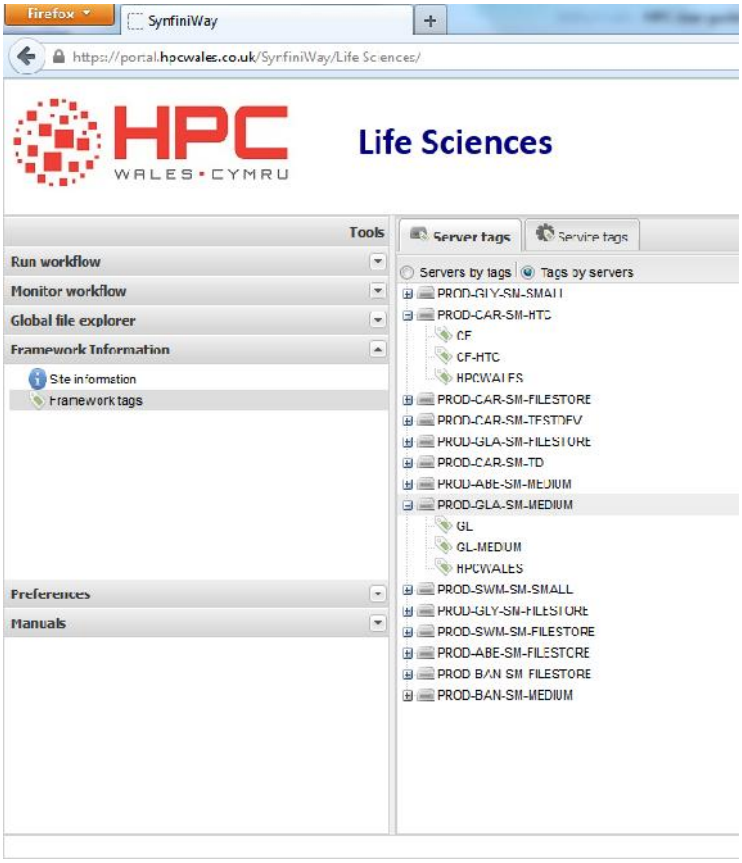
points are likely to be created for particular entities, e.g., projects, teams, disciplines. Such mount points will be expressly controlled by RBAC.

All SynfiniWay mount points are accessed from within that same single explorer view. The main file operations supported are:

- Create new file or directory.
- Delete file or directory.
- Copy and paste file or directory. Items can be moved between any pair of mount points, within the same site and between sites, as a single operation.
- Upload and download files between the desktop and any mount point. This transfer operates as a single action directly to any remote file-system. There is no staging on the web server location.

9.3.5 Tools: Framework information

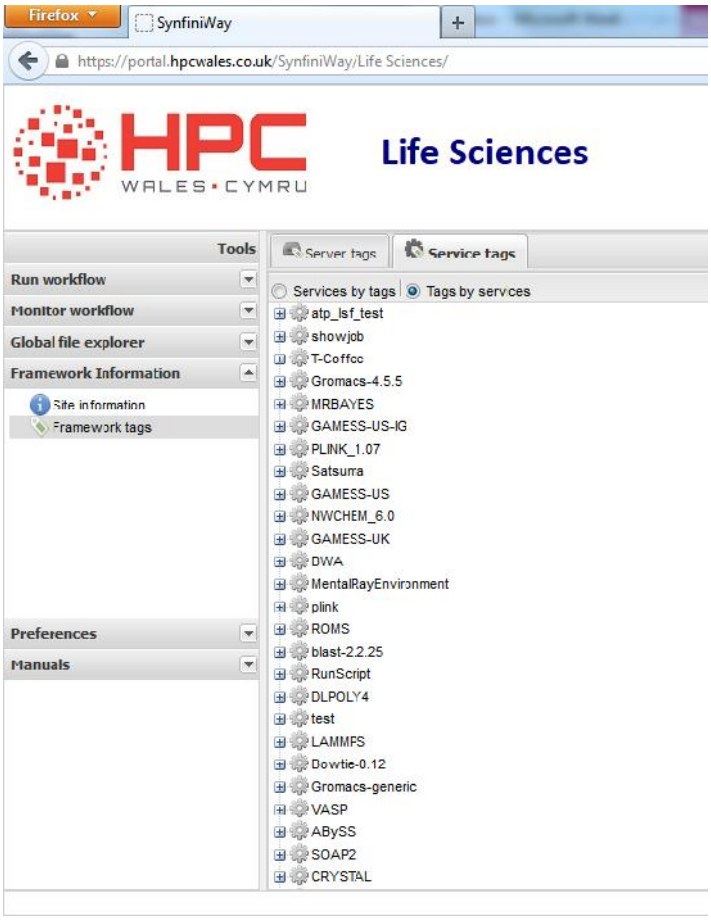
Server tags are arbitrary labels associated with SynfiniWay Service Managers (SM). The administrator sets these tags as an aid for the end-user to select the correct service execution location. When the meta-scheduler is determining the placement of a particular service, any server tags set by the user will be compared with those defined on the individual SM's where the service is available. The user will enter the value of the tag into a variable of the workflow.



Not all workflows will have such a variable defined. In this case the user has no explicit control over the location where tasks of the workflow will execute. This is generally the default situation when services are developed since tasks are expected to potentially run on any of the HPC Wales sub-systems, wherever the application or function for the task is supported. In such cases the user does not need to be concerned with the sub-system selected by the meta-scheduler since all data movement for the workflow is entirely managed by the SynfiniWay workflow engine. Equally, files are accessible from the global file explorer wherever they are located across the HPC Wales network.

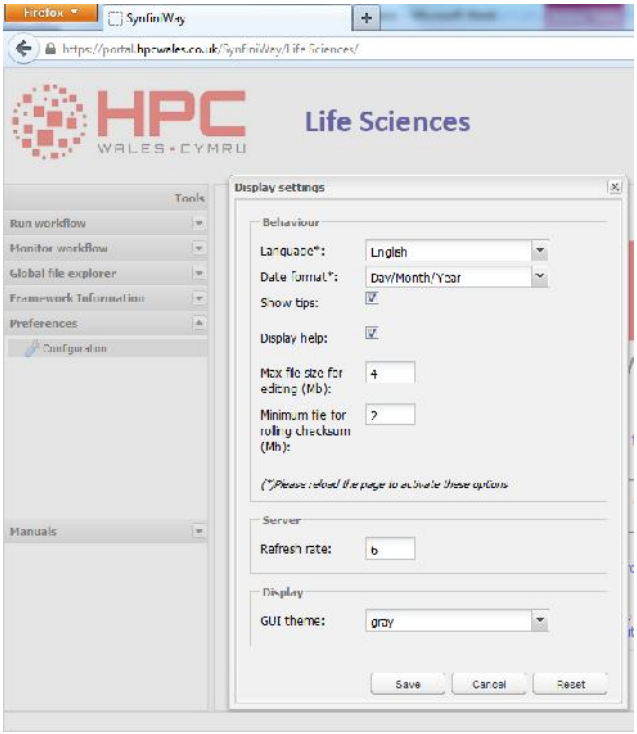
Another type of arbitrary label is the *Service tag* associated with each published service. Such tags are employed to represent an attribute of the service, as compared to previous tags for the servers comprising the SynfiniWay framework. They are similarly used by the meta-scheduler to limit selection of service to those that have that specified tag. The user will enter the value of the tag into a variable of the workflow.

The *Service tags* panel will list all services available across the entire HPC Wales framework. No tags are yet defined for any of the services.



9.3.6 Tools: Preferences

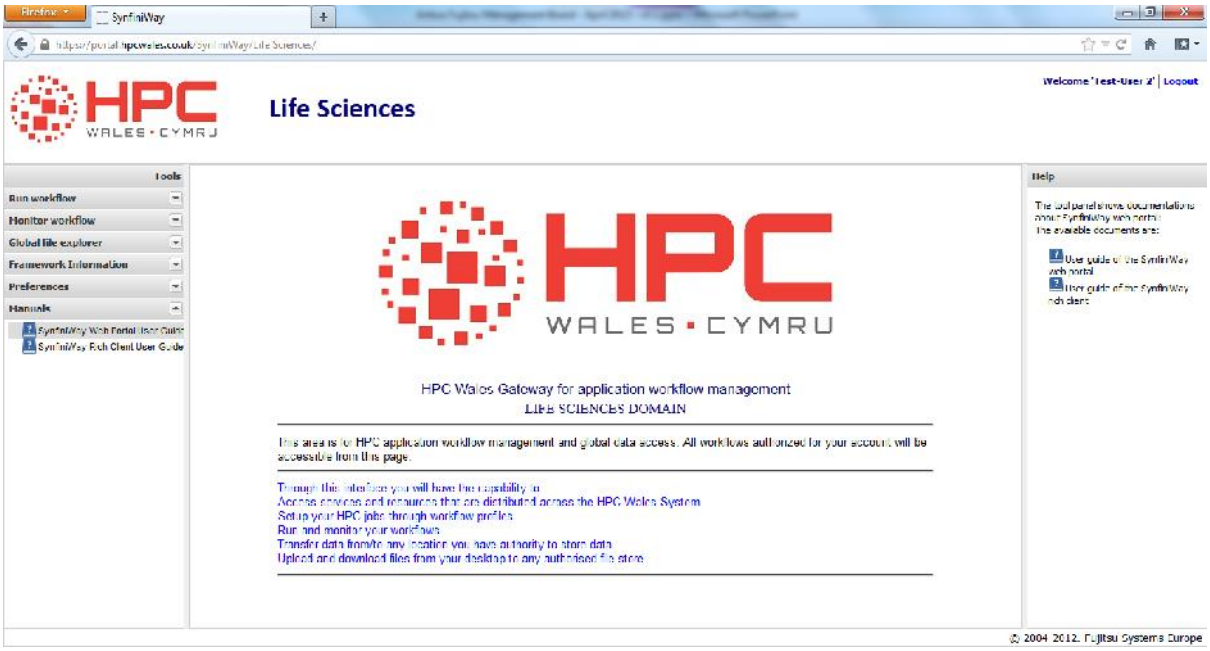
The Preferences tool provides access to a limited set of configuration parameters for the browser display that are selectable by the end-user. Any other customisation for the web interface is done by the administrator.



The available settings may expand in future versions. Equally the look and feel as described in this guide may differ in certain aspects (e.g., colour, font) to the actual deployed portal.

9.3.7 Tools: Manuals

The Manuals tools provide a link to selected standard SynfiniWay product manuals. These guides are those that are most applicable to the end-user of SynfiniWay.



When clicking on the documents in this area a new tab will be opened in the browser containing the selected file. Other documents may be added to this tool area.

9.3.8 Leaving SynfiniWay

At any time it is possible to return to the associated gateway home page by clicking the back button of the web browser.

Alternatively, clicking **logout** in the top panel will exit the SynfiniWay portal.

9.4 Using SynfiniWay Workflows

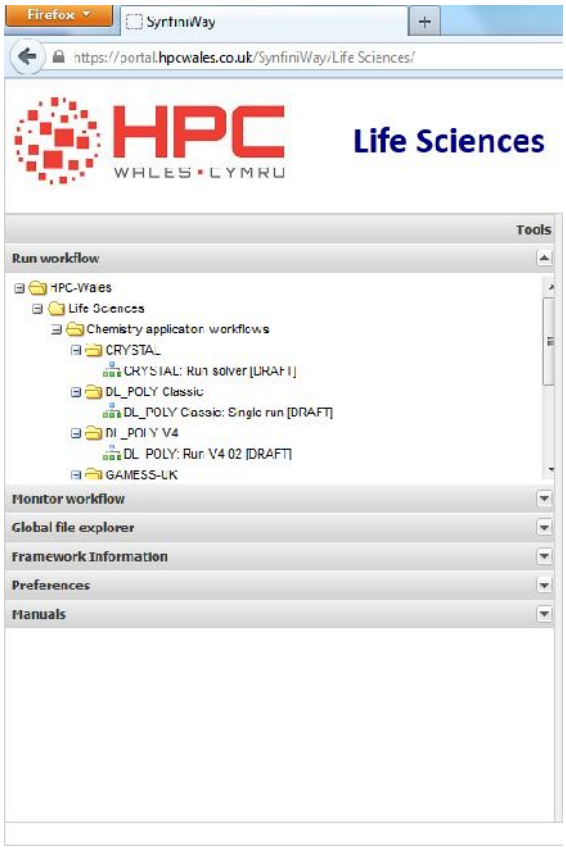
9.4.1 Introduction

This section describes the use of SynfiniWay workflows through the web browser interface.

9.4.2 Selecting workflow to use

Permissions are managed using a global RBAC mechanism linked to your HPC Wales login identity. These permissions are arranged in role groups that are mapped directly from the group defined in the HPC Wales Active Directory user management system. In the following sections we will use examples from an identity in the **Life_Sciences_Users** group.

Select the workflow to be used for your study by navigating through the folders and list of available workflows. To aid your selection the workflows are categorised by convention into folders according to a specific attribute. Current attributes that are used are application name or computational domain. Other attributes may be used in future, such as project name.



Your selection of workflow to run will be based on several criteria, including:

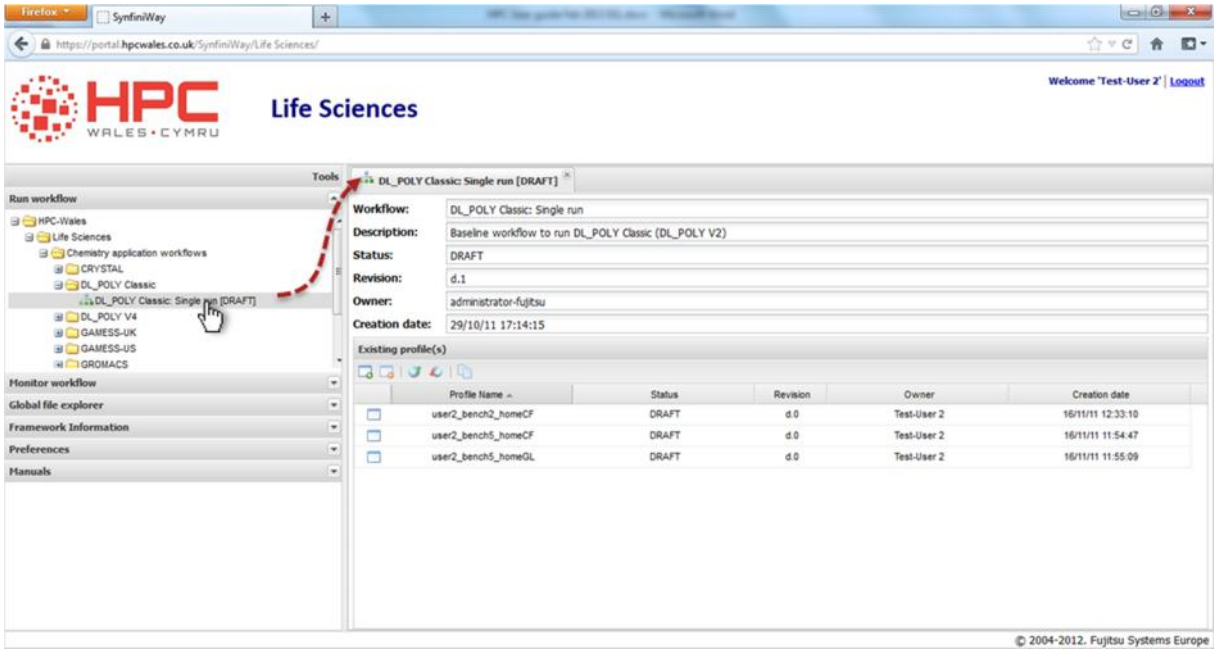
- Main core application you want to use;
- Application version, where identified as a separate workflow;
- Structure of the complete process around this application;
- Publication status (i.e., DRAFT or PUBLISHED)

A convention has been adopted at HPC Wales to identify workflows in the gateway using a name that includes the main application name as a label indicating the workflow process. However, this approach may be adapted or changed

depending on the developer of the workflow.

Details about the function of the workflow will generally be provided in one or more forms. A string in the description field of opened workflow will provide a succinct description of the overall process (see sub-section 9.4.4). Eventually more extensive descriptions should be found as an associated guide located in the document management area of the Gateway. Where further information is required contact the workflow owner (see sub-section 9.4.4) or the HPC Wales administration team.

After selecting the workflow you will need to define inputs for your specific job. Left-click once on the selected workflow label to open the list of input forms in the centre panel of the gateway web page.



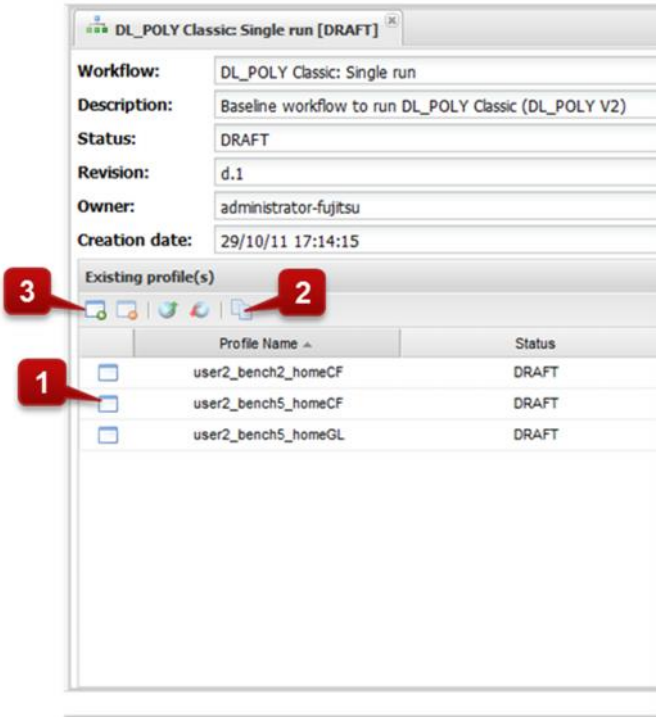
Note that in the following images the help panel has been removed. This is set in the **Preferences** tool.

9.4.3 Using workflow profiles

For each workflow the input parameters are defined in a form referred to as a *workflow profile.* A workflow profile is an entity that stores the consistent set of parameters – options and file definitions – for a given process. Multiple profiles can be created for each workflow. There are three approaches to setting up workflow profiles:

1. Replace values and save in an existing profile.
2. Create a new completed profile using **Copy profile as draft** command on an existing profile.
3. Generate a new empty profile with the **Create new Profile** command.

A right-click on the profile name reveals the same options given in the profile menu bar.



Each individual can choose which approach to adopt in managing their workflow profiles. The following table gives suggestions of the considerations for each approach.

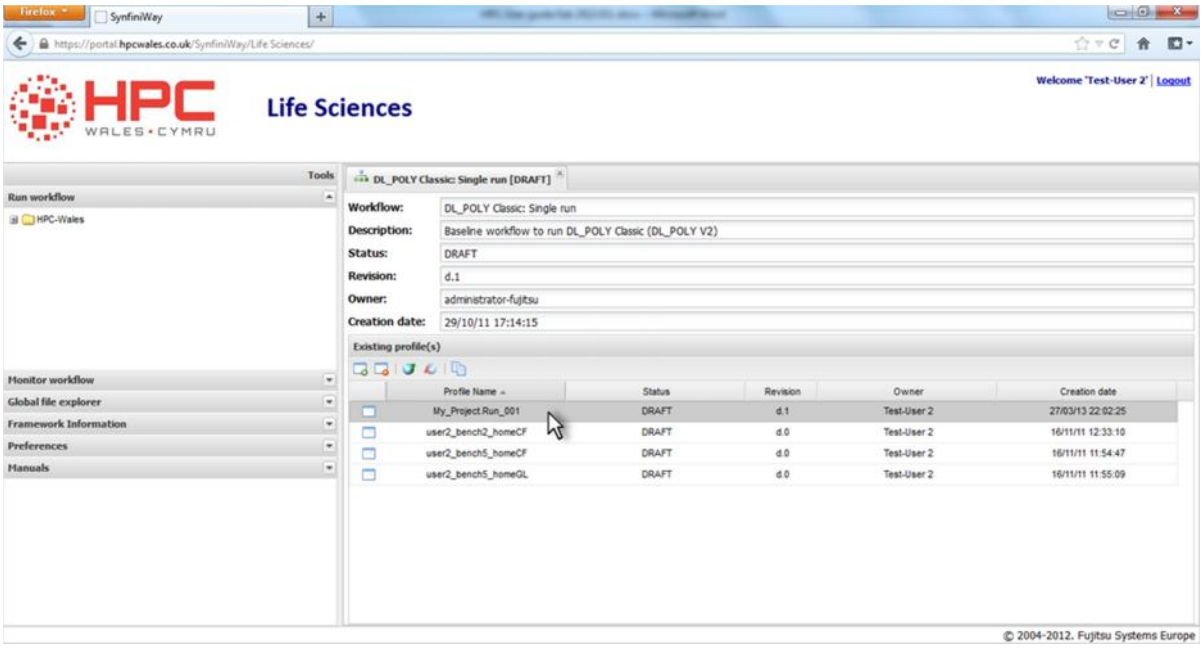
Approach	Usage
Replace and save	This is the quickest approach when only minor changes to parameters are required, e.g., processor count, but lacks any tracking. Most likely to be adopted where several profiles are defined for different scenarios in the same workflow, e.g., • Single/Double-precision • Short/Long run based on model values (cycles, cells) • Short/long run based on loop iteration count
Copy profile as draft	Using new profiles allows for better tracking. This approach retrieves earlier parameters that are progressively updated for different option settings or input files. This approach is commonly the preferred methodology for systematic analyses within a given project.
Create new profile	Where there is little or no overlap between different runs of the workflow the appropriate choice is to simply create a new profile. The variables to be defined are listed in the same way within the profile. Variables that do not have a default value will have an empty settings field. The best practice approach for workflow variables is to have a drop-down list of fixed values that the user selects using point-and-click. As workflows evolve it is expected that the span of selectable variables will increase so that creating new profiles

	requires little additional effort from the user.

Depending on the RBAC definitions you may be able to view the profiles of other users. This is useful for disseminating a demonstration setup, as illustrated in the previous image, or to share the tests being made across a team. With the baseline RBAC permission you will be able to run the profiles defined by other users, but will not be able to modify them. Such jobs may therefore fail if commensurate access is not allowed to the input or output files.

9.4.4 Defining workflow inputs

In the following example one of the existing demo profiles was copied as a draft which is named **My_Project.Run_001**.



Double-click on the workflow profile to open the parameter input forms. These are presented in three separate tabs:

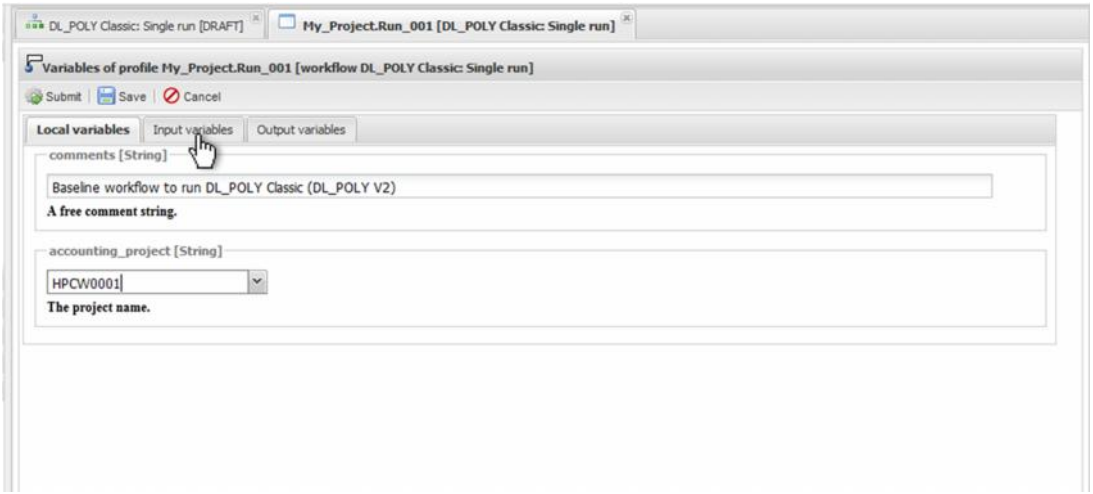
- **Local variables** are used primarily for descriptive fields and parameters that apply to the overall workflow environment. These also include variables predefined by the SynfiniWay system.
- **Input variables** are used for command options, logic control and input files to the process to run the given simulation.
- **Output variables** are mainly used to set the location for files returned by the workflow.

There are two pre-defined variables for each workflow.

- The **comments** variable is used to provide a succinct description of the purpose of this workflow profile.
- The **accounting_project** contains the project code for this job. It is a compulsory parameter, and the workflow will abort if it is not defined. The list of accounting

project codes you can use is obtained from the HPC Wales system administration team.

Other local variables may be defined as needed during the development of the workflow.



To switch to other variable forms click on the relevant tab.

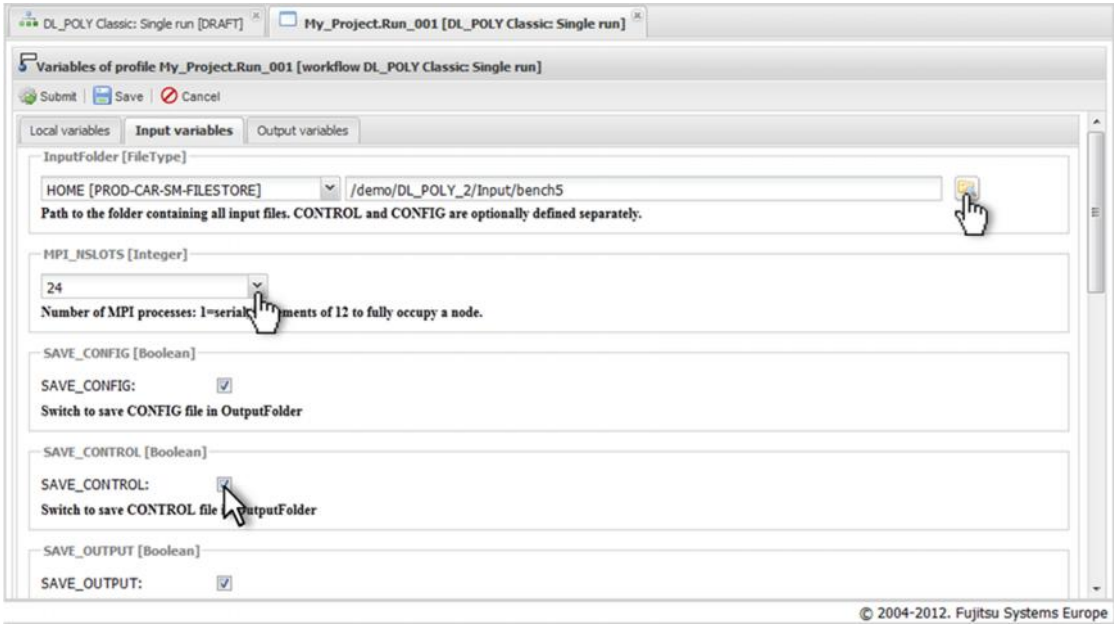
Most parameters for the selected workflow profile will be set in the Input variables form. Variables may have different types: byte, Boolean, char, short, int, long, float, double, String, Host and File:

- The (byte to String) types have the same definition as in the Java language,
- The Host type is used to hold a reference (ID) of a SynfiniWay server. The Host ID is created as a string type variable,
- The File type is used to describe a SynfiniWay file. This type is a complex type formed by combining a Host type and two string types.
- An input variable can also be of type array of <basic_type>.

For each variable the display will show:

- **Variable name**. This is the single string used throughout the workflow and service logic to contain the relevant value.
- **Variable type**. One of the types stated above.
- **Comment**. A statement provided by the workflow editor to describe the purpose of the variable and its use.
- **Input field**. The value of the variable that will be passed to the workflow.

The following example demonstrates the use of files, integer and Boolean variables. In each case the value can be selected using point-and-click control. Clicking on the drop-down list for an integer, such as the MPI_NSLOTS variable, will display the values proposed by the workflow editor. For example, this variables offers the choice of 12 (default), 24, 48 and 72 processes (cores). A Boolean variable is toggled through a tick-box.



The locations of input files are easily added to the variable list use the global file explorer available from the icon within each variable box. Usage of this explorer is described in section 9.5. For the example above the explorer is used to locate and select a folder for the first variable (**InputFolder**) and individual files for the second and third variables.

The description of the file or folder in the input and output variable forms has two components:

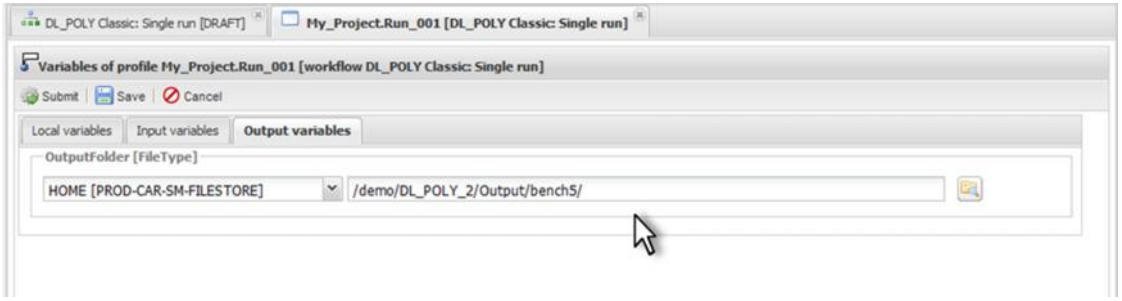
a) Selected SynfiniWay virtual mount point.

b) Path beneath this mount point.

SynfiniWay virtual mount points are described in section 9.5. For the definition of file variables the mount point can be selected separately using the drop down list. The path can then be typed manually into the field provided. However, the recommended method is to select both components using the file explorer.

Output variables usually require a definition of the location where the workflow should send the resulting individual files, or a complete folder containing multiple files. As for the inputs, selection of this location can be done using the global file explorer. Conversely, the selection has differs to inputs in two respects:

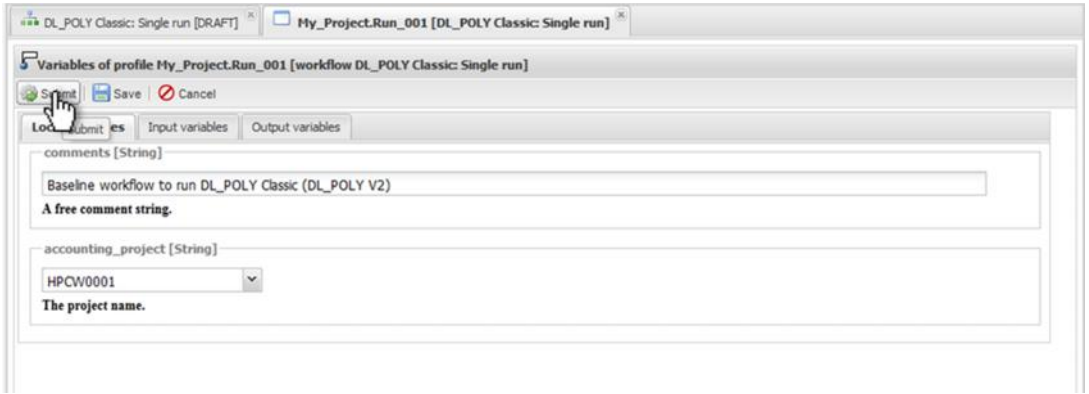
- If a folder is defined as an output variable then this will be the location where the output file will be sent. A “/” character may need to be added manually to complete the path component definition. The workflow will create any further sub-folders beneath this location as required.
- If a file is defined as an output variable then this name must correspond precisely to a file located within the run directory of the relevant workflow task. This name will be manually entered into the path field since it is presumed that the named file does not already exist within the file system, and so cannot be located using the file explorer. Wild cards can be used to select multiple files. These are defined using the Java regular expression syntax.



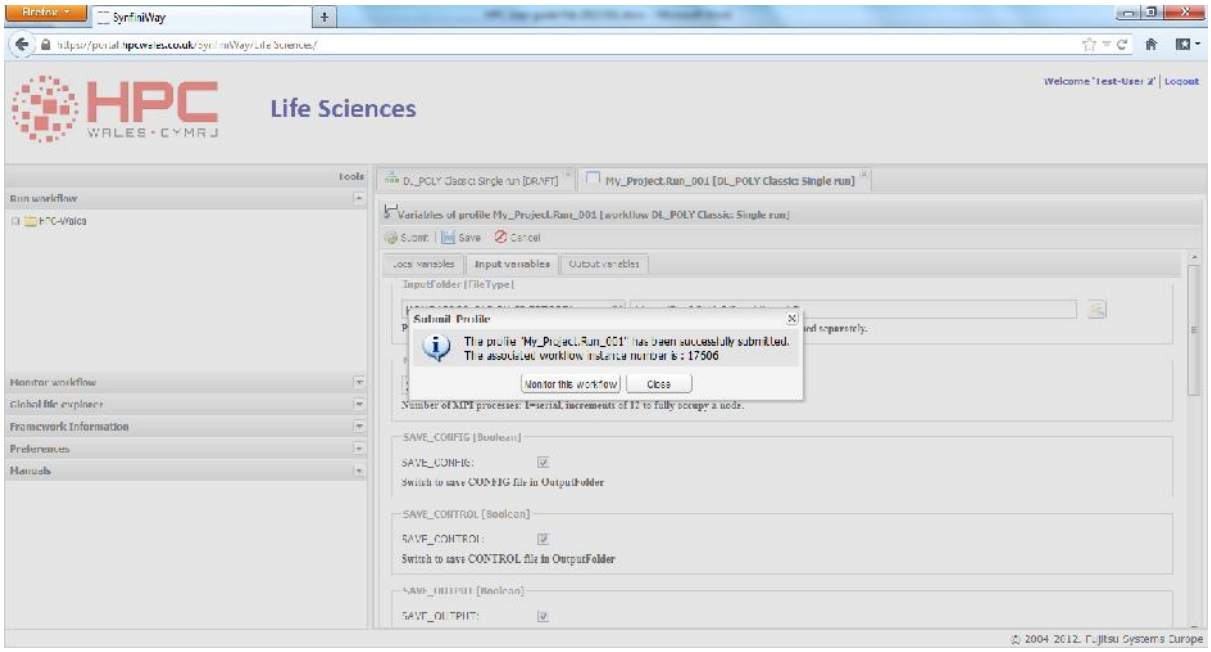
After all variables are defined the workflow profile must be saved prior to submission.

9.4.5 Submitting workflows

A saved workflow profile is submitted by clicking on the **Submit** button. Note that if there were any updates to the profile then a **Save** must have been done previously.



A successful workflow submission will be acknowledged by a message that includes the workflow identifier. This identifier should be used to track the workflow progress in the **Monitor** tool.

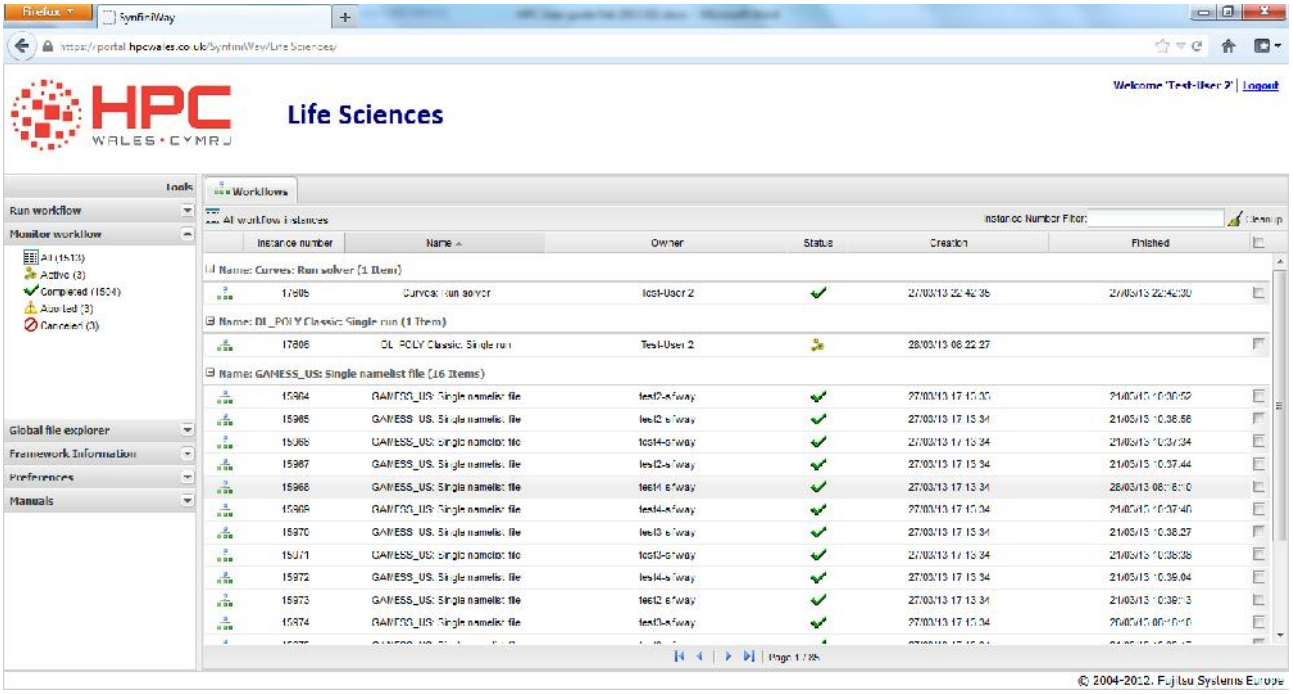


Clicking the **Monitor this workflow** button in this pop-up message provides a shortcut to track the workflow.

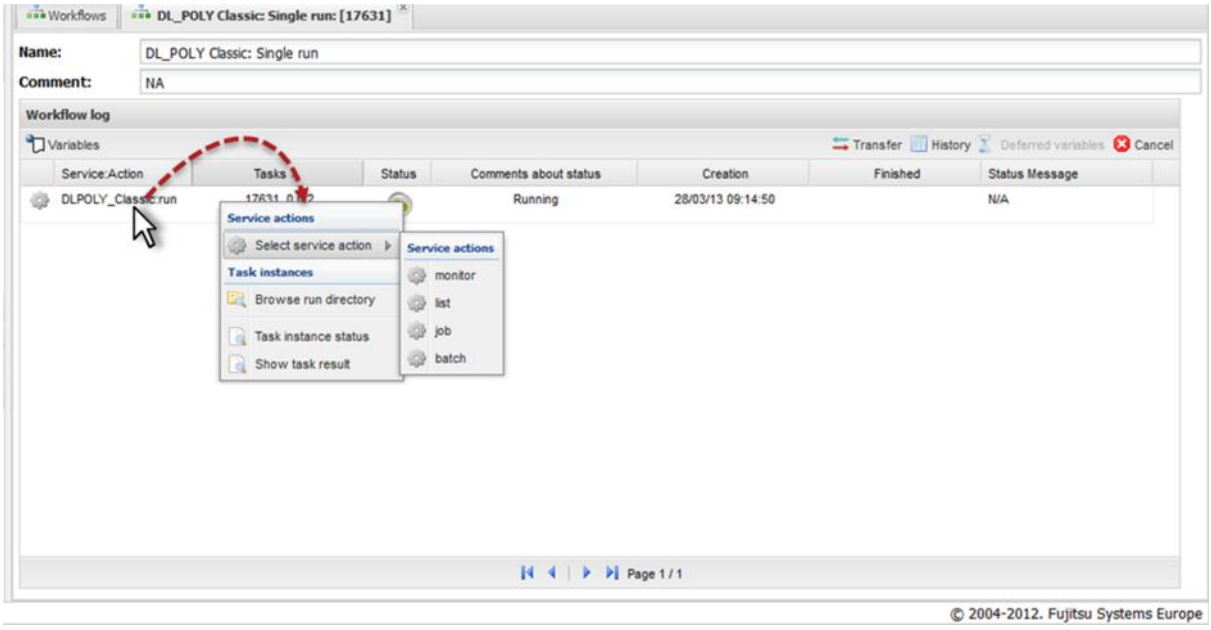
9.4.6 Track workflow state

After a workflow is submitted an entry will appear in the **Monitor workflow** area identified by its instance number. Initially the monitor view will show all workflows that have not been cleaned, either manually or automatically. Workflows in each of the four states can also be viewed separately by selecting the relevant list in the monitor workflow tool panel.

In the following view of **All** workflows from the Life Sciences gateway three workflows are in the **Active** state. If there are more current workflow instances than can be displayed in one view then you will need to switch between views using the start/left/right/end controls at the bottom of the centre panel. Workflow entries are grouped by their name.



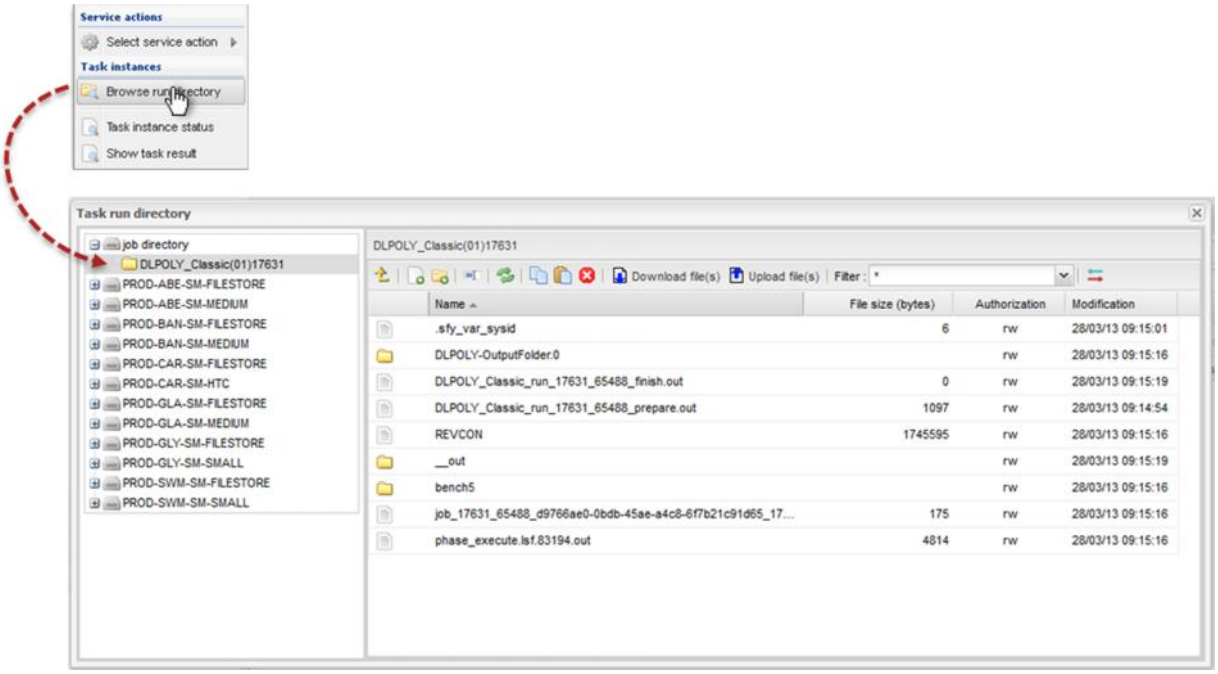
Next, to obtain more detailed information on a workflow double-click its entry in one of these lists. This action will open a new tab in the central panel that shows the status of all individual services within the workflow process. Only the services that are running or completed are listed. Services yet to be dispatched are not visible in the list view.



Several actions can be taken from this view to obtain more detail on the service. This information is accessible for all states of the service by a right-click on the service name to reveal the selectable menu.

- Browse run directory opens a file explorer centred on the working directory of that task.
- Task instance status provides system information about the task.
- Show task **result** displays informative messages defined within components of the application service.
- Select service **action** will show the list of application-specific monitors that can be run.

9.4.7 Reviewing work files

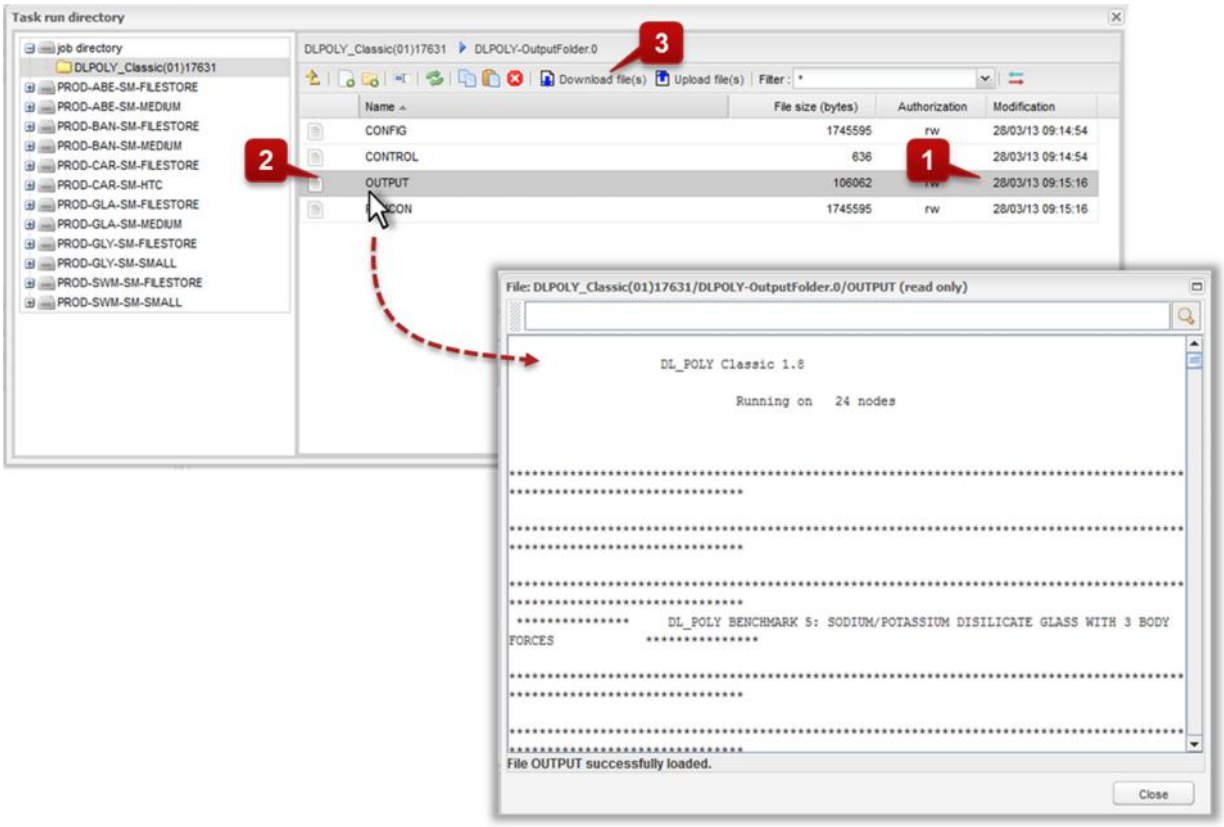


SynfiniWay creates a unique temporary working directory for each task it runs. Navigation to this directory is generally not possible or practical using global file explorer alone. Instead direct access to this working directory is provided through the Browse run directory function. This will open a file explorer window with the run directory highlighted. Other directories will also be shown corresponding to all authorised global mount points as shown in a standard file explorer view.

You can navigate below this task directory in the usual way (see section 9.5 on the use of the file explorer tool). This function can be used while the task is running or after it has finished, whether normal or abnormal completion.

Common actions within this view include:

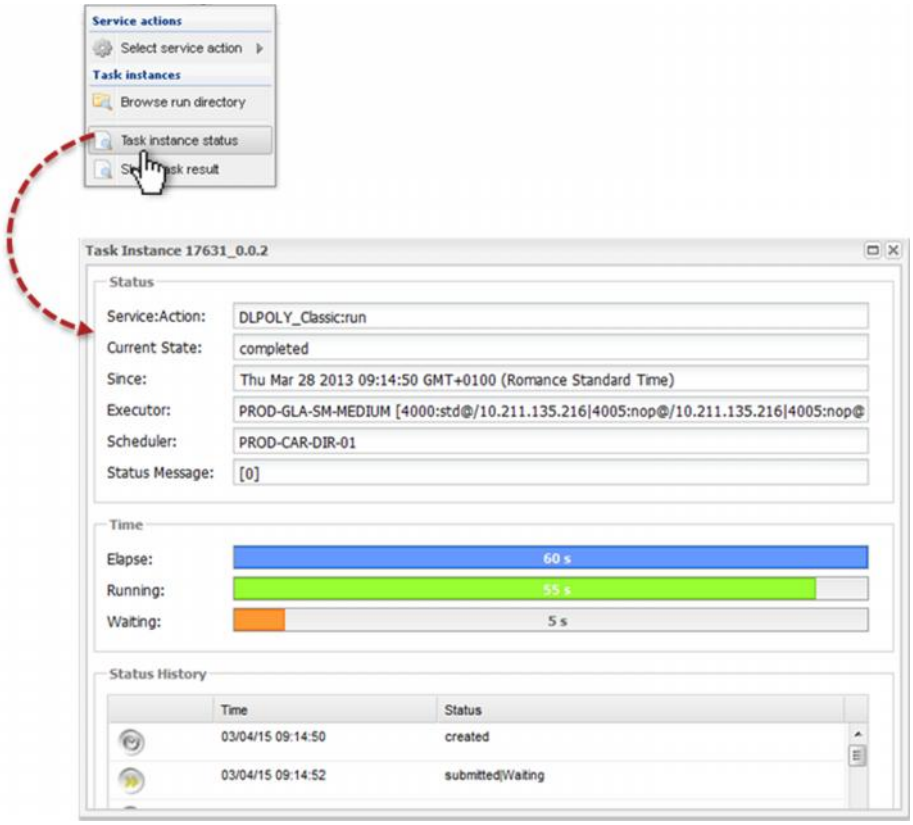
1. Track output file sizes while the task is running. Allows you to observe that the application is progressing, that the expected output files are being generated and the size is evolving normally.
2. Clicking on a file name will open the content within an in-built text editor. This initial inspection will indicate if the simulation is giving acceptable results, that the simulation is moving forward in time, etc.
3. Download a selected file for analysis on the desktop.



Accessing files directly in this way allows you to review any file within the work directory. Service sub-actions provide another form of monitoring for faster access to file content, but also to apply other utilities to filter and report on progress.

9.4.8 Checking system information

The system-related status of a selected task can be checked by selecting **Task instance status** from the action menu. This will reveal a new panel that display a range of information that track the progress of this task through the SynfiniWay system.



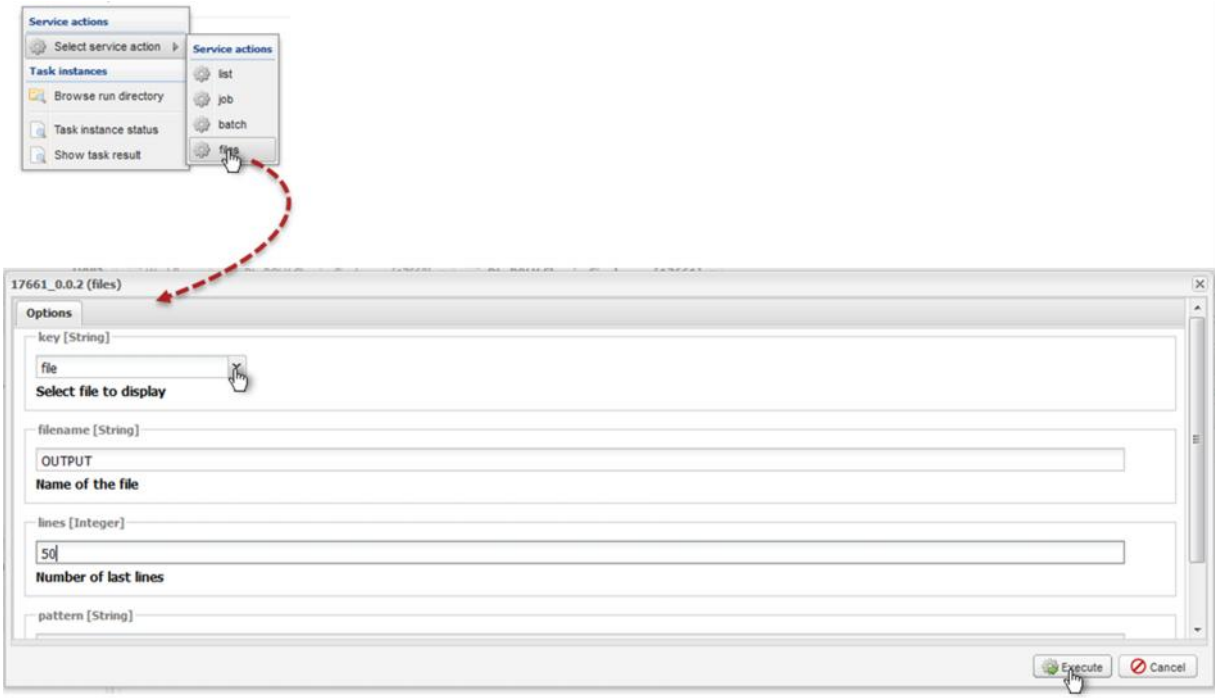
Key elements of interest from this table include:

- Definition of the service and action run by the task. Validate that this is consistent with the process selected for your analysis.
- The Executor shows from which SynfiniWay server this task was run. The name of the server generally indicates the site and compute server type that was used. In the above example PROD-GLA-SM-MEDIUM shows that the site was Glamorgan (**GLA**), and the compute server was the **Medium** cluster. The HPC Wales system administrators can provide details on these code names. All available servers are listed under the **Framework Information** tool.
- Time information refers to the occupancy of the task within the SynfiniWay system. Definitive statements of the CPU and elapsed times within a particular compute server are accessible through the consolidated job statistics database and through other service sub-actions.
- Check the **Status History** to track the overall progress of the task through the different stages of execution. Any error in submission or execution will be indicated here.

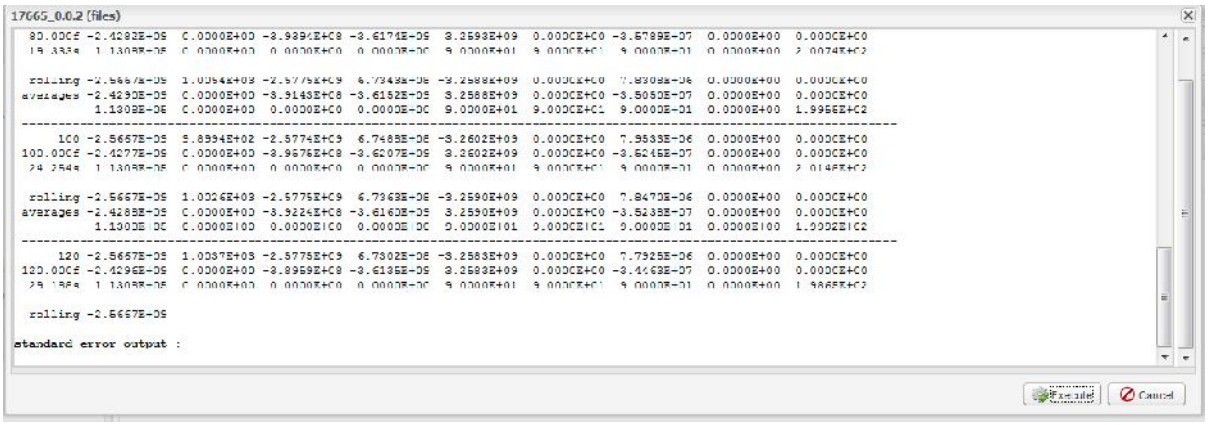
9.4.9 Running application monitors

SynfiniWay provides a capability to run one or more **Service actions** related to a target service task. These actions can be launched while the task is running or finished. The list of actions is not fixed and is application-specific. Monitors will generally be used to obtain information from the running application. But some actions can also be assigned to system-related information, such as enquiries on the batch resource manager.

Most initial workflows deployed will offer relatively simple monitors to allow direct access to some of the key numerical information for each application. Such basic monitors provide a way to rapidly and easily inspect sections of the main output files. The following image shows the selection of a monitor to view the last lines in one of a set of files generated by the specific application. In this example we want to view the last 50 lines of the file named OUTPUT, one of the named files produced by the DL_POLY Classic application.



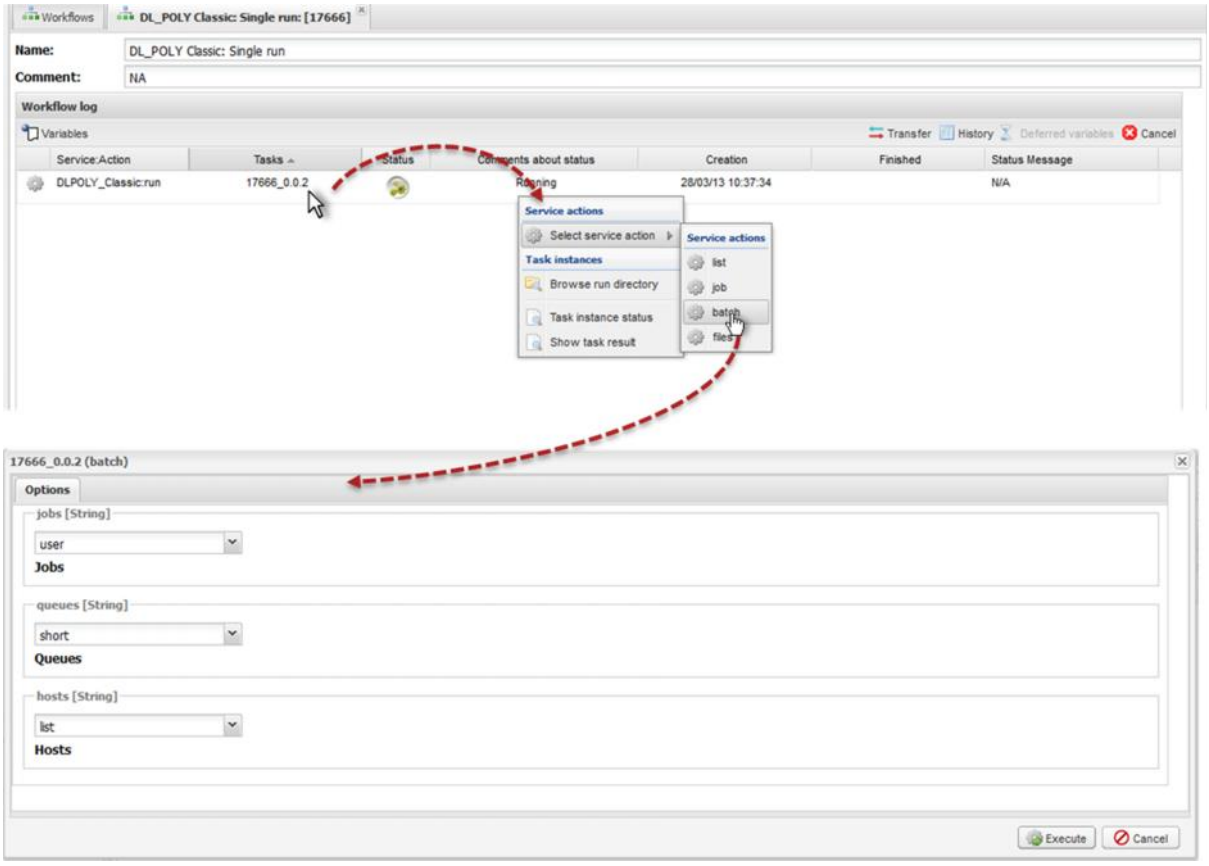
If the file exists while the task is running the monitor will capture the most recent lines added.



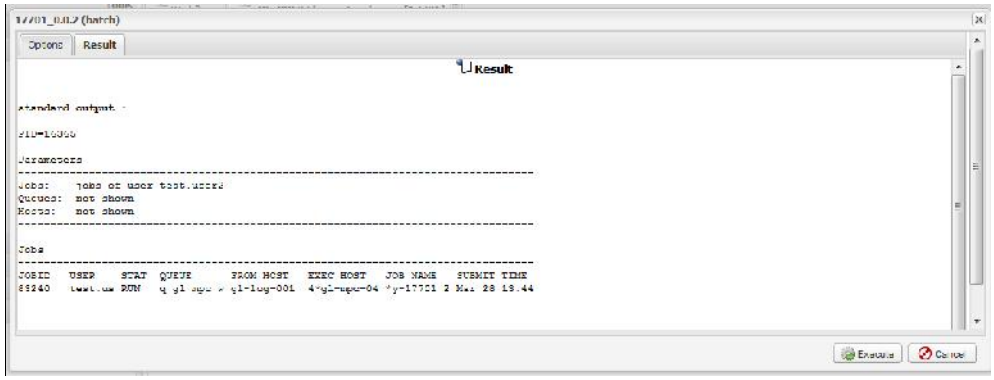
After the task has completed the result will be the terminating lines in the monitored file.

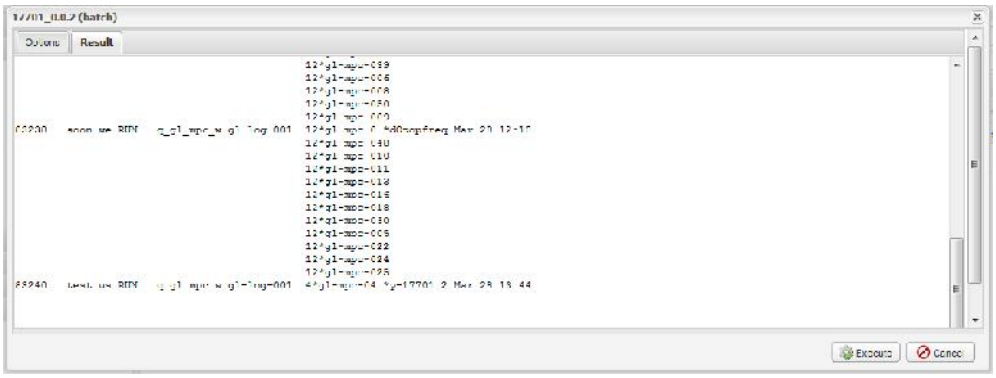
These monitors can be enhanced and expanded by HPC Wales according to the user requirements. It is expected that such monitors will eventually support a richer set of capabilities, such as application run-time control (stop, checkpoint), table or 2D image generation, data filters and dynamic summary reports.

For each workflow task the **batch** action allows you to check three types of run-time information from the batch resource manager: jobs, queues and hosts. To deactivate any of these outputs select “none” from the sub-option list.

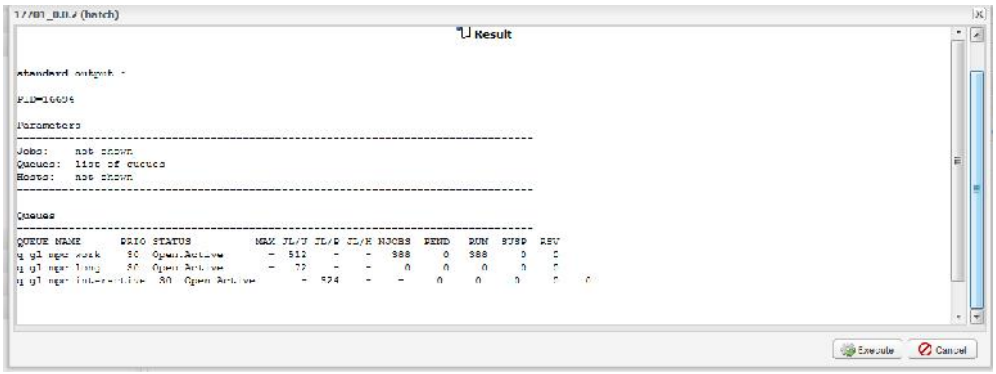


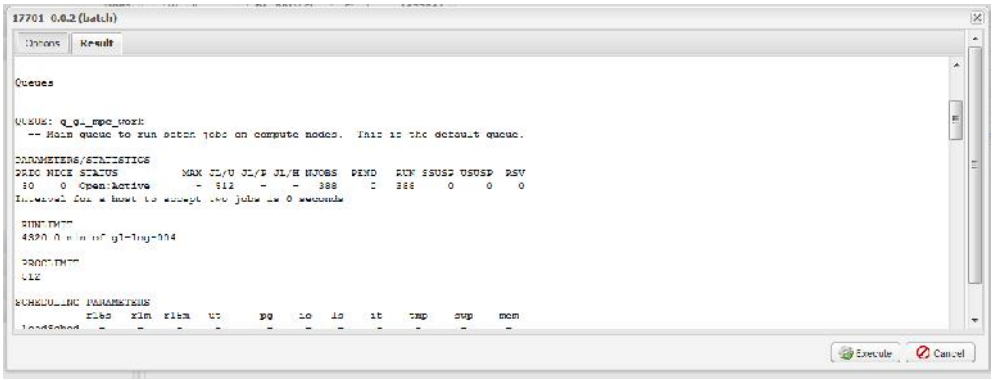
The job option allows you to check just your jobs (*user*) or all jobs (all) within the resource manager queues.



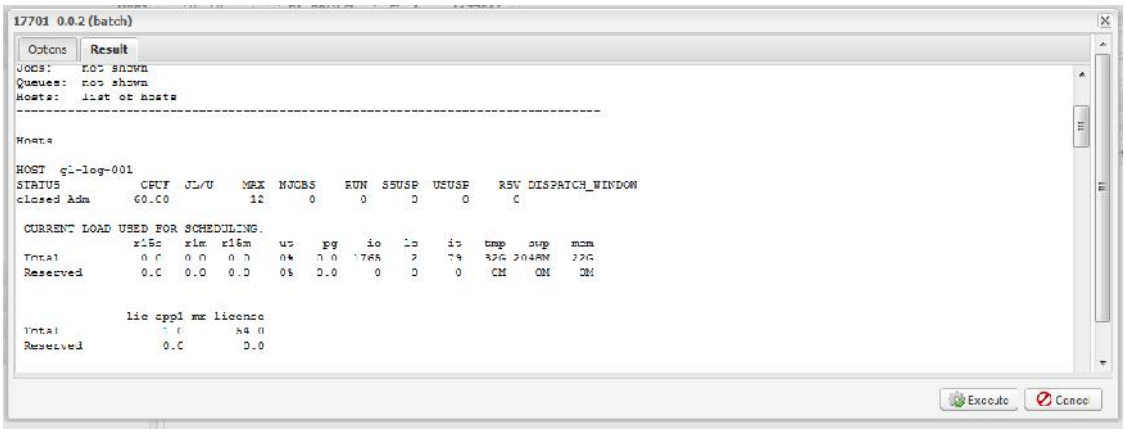


The **queue**s option provides information on the resource manager queues in short or long format.





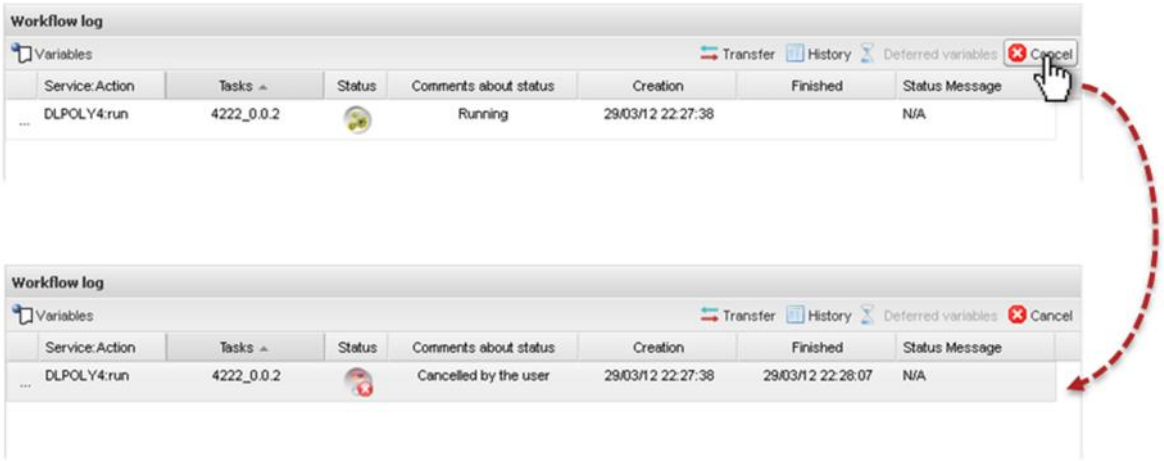
The **hosts** option give a list of the host information from the batch resource manager at the time of the job execution.



Finally, the **job** action displays the detailed information on this specific task in the batch system.

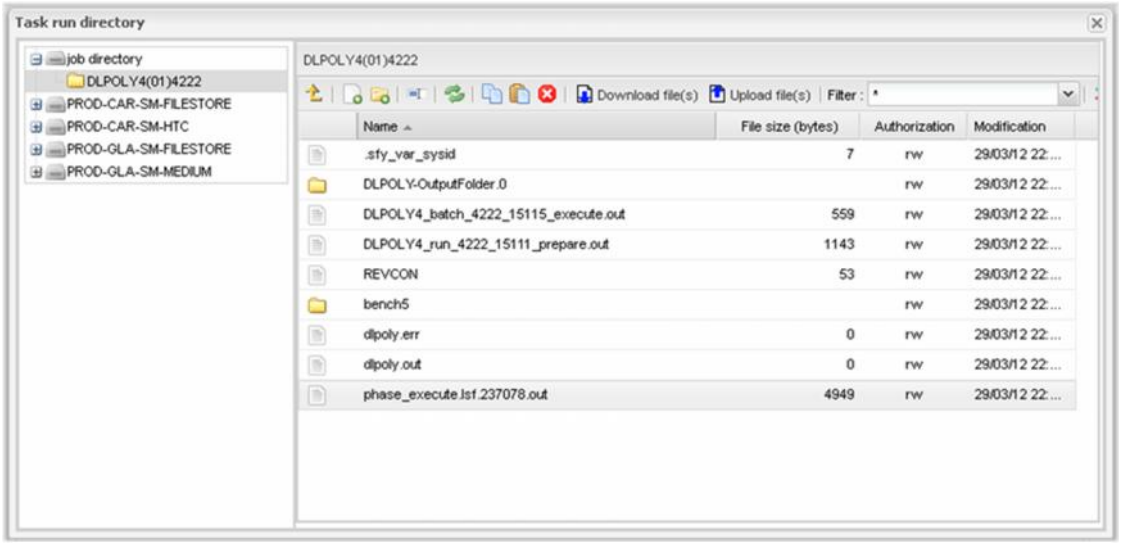
9.4.10 Stopping workflow

Under some circumstances you may want to stop a workflow that is running. For instance, after reviewing the file output being generated it may be clear the simulation is not progressing correctly. Stopping a workflow is done by clicking on the **Cancel** button of the workflow instance panel.



If the running task or tasks involve a batch job running then this action will kill the running batch job associated with each workflow task.

Access to the run directory remains possible until the workflow is cleaned.



9.4.11 Cleaning workflows

When a workflow executes a task on a Service Manager it uses a storage area known as the job directory to store input files, output files, and temporary files created during the execution of the task. Each task has its own job directory, and for certain workflows these can be created on multiple Service Managers. These directories with their files remain on each SM used during the workflow execution until the workflow is cleaned up.

SynfiniWay provides a workflow option that enables automatic clean-up of these directories and files and also the information about the workflow itself (such as the complete status of workflow task execution). Clean-up is generally specified to be automatically performed at a reasonable time, for instance 30 days, after the end of the workflow. This delay is set by the workflow editor or framework administrator. The actual clean-up delay will be adjusted as the utilisation of the framework evolves. Note that this system clean-up action will take effect without any notification to the owner of that workflow instance.

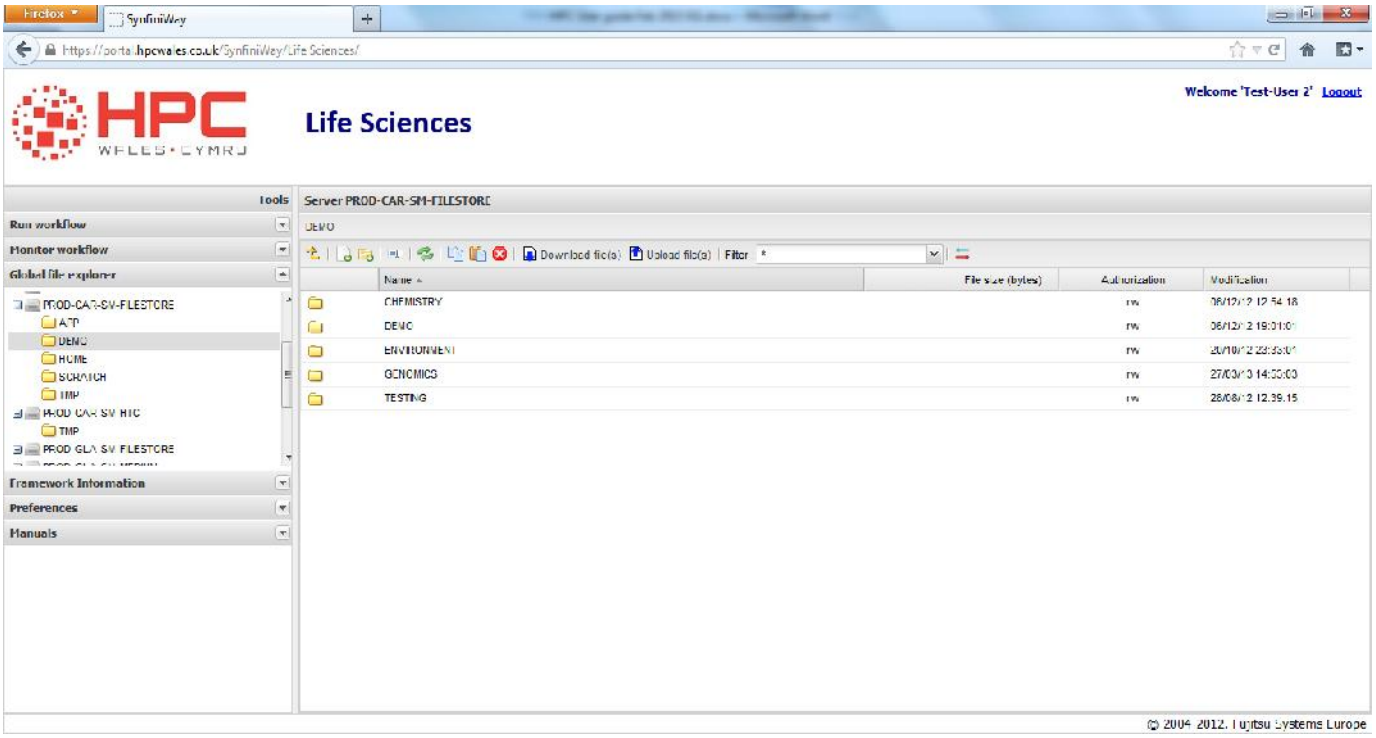
If the automatic clean-up option is not set, the user will have to manually clean-up the workflow. HPC Wales administrators will define a convention for when to clean-up workflows based on overall resource usage. Users should broadly consider the following steps before cleaning a workflow. In any case note that only the temporary objects created by SynfiniWay during execution are removed at clean-up. Workflows are expected to include at least one explicit file output point so that the relevant data is implicitly copied to a designated location.

- Review the overall workflow status: active, completed, cancelled or aborted.
- Check the system status of individual tasks.
- Inspect the log output from the workflow and individual tasks.
- Ensure that all required result files have completely transferred back the target location.
- Issue a clean-up of the workflow once you are satisfied that all necessary information has been obtained and that the completed workflow instance will serve no further purpose.

9.5 Using the Data Explorer

9.5.1 Navigating file systems

Files located on any HPC Wales sub-system across the distributed infrastructure are accessible through a single view using the **Global file explorer** tool.



Filesystems are referenced through a top-level SynfiniWay virtual mount point. This object can be mapped to any directory with the target filesystem. Multiple virtual mount points can be defined with the same filesystem. Navigation is only possible below this mount point; no files are visible above this point.

In the baseline HPC Wales environment such virtual mount points have been defined to map to selected higher-level directories. As the solution evolves other mount points may be defined by the system administrators to provide specific mount points for more representative objects, i.e., domains or disciplines, groups or teams, projects. Currently one such mount point, named **DEMO**, exists on the Cardiff site.

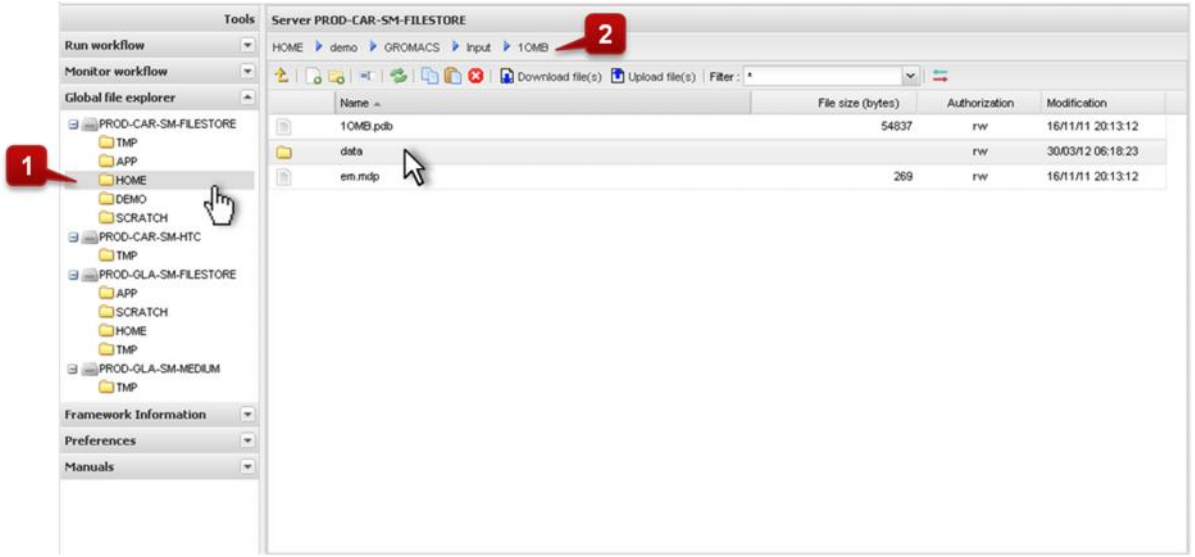
Virtual mount points are quickly created and immediately accessible to users of the framework everywhere. Moreover they are controlled by the SynfiniWay RBAC mechanism for easier management of group permissions.

Begin navigation by selecting the virtual mount point in the left panel. Mount points are defined only for SynfiniWay Service Managers (SM). Two types of SMs are deployed; one for compute activity associated with different cluster types, and one to serve data from each site. The baseline mount point convention is given in the following table. Only **HOME** does not map to a top-level filesystem directory. Instead this mount point represents the individual home area for the login account on each designated host site.

SM type	SM logical name	Mount point label	Filesystem directory
Compute	PROD-<SITE>-SM-<SUBSYSTEM>	TMP	/tmp
Data	PROD-<SITE>-SM-FILESTORE	TMP	/tmp
		APP	/app
		HOME	/home/<account>
		SCRATCH	/scratch

To locate and access files there are two basic steps:

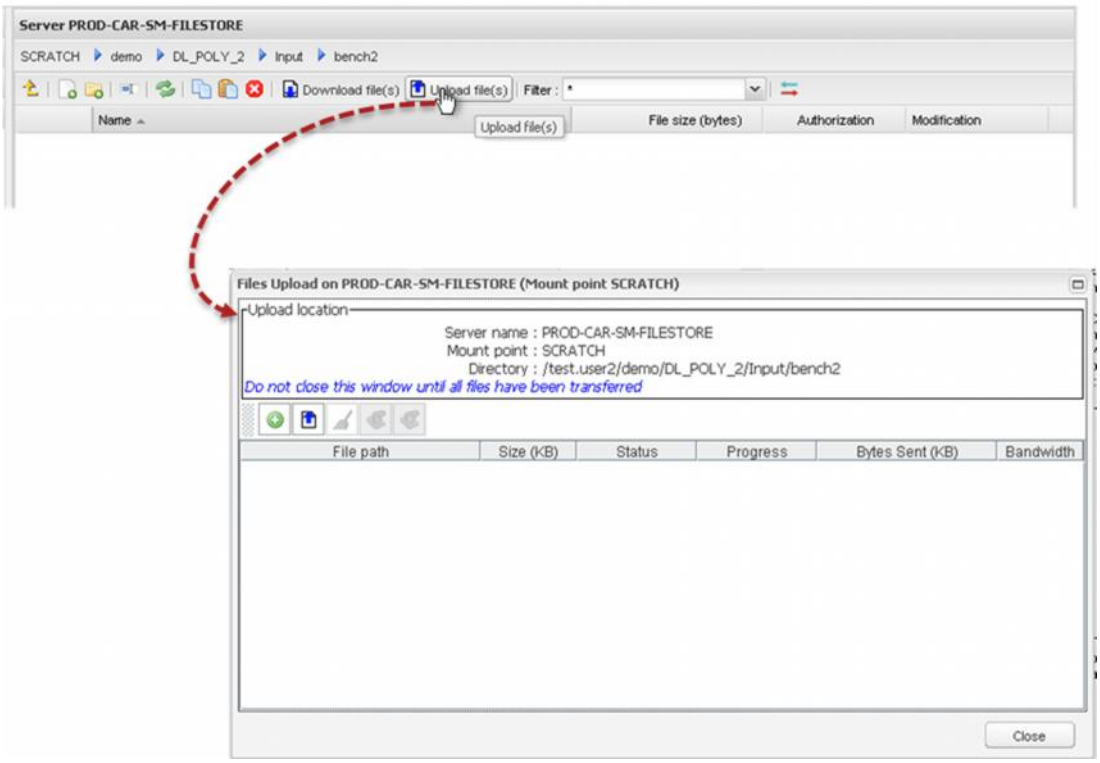
- Unrolling the SM logical names and selecting a virtual mount point will open that directory content within the centre panel of the browser window.
- From this area you can navigate as usual by double clicking on directory or folder names.



9.5.2 Uploading files

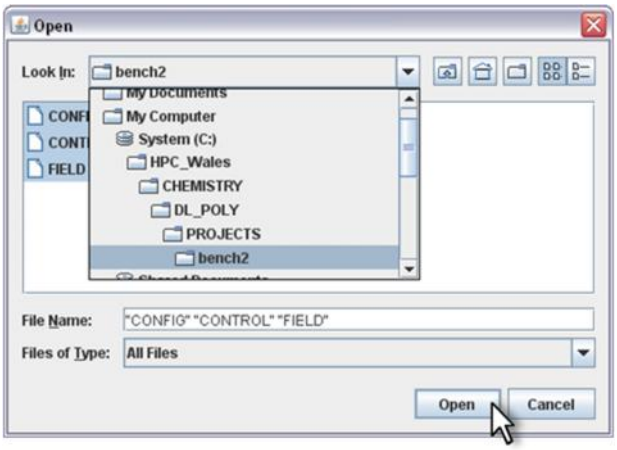
Navigate to the location where you want to place the files or folder being uploaded from the desktop. Single files, multiple files and folders can be uploaded as a single action.

In the first example a directory is first created to receive multiple files (see subsection 9.5.5).

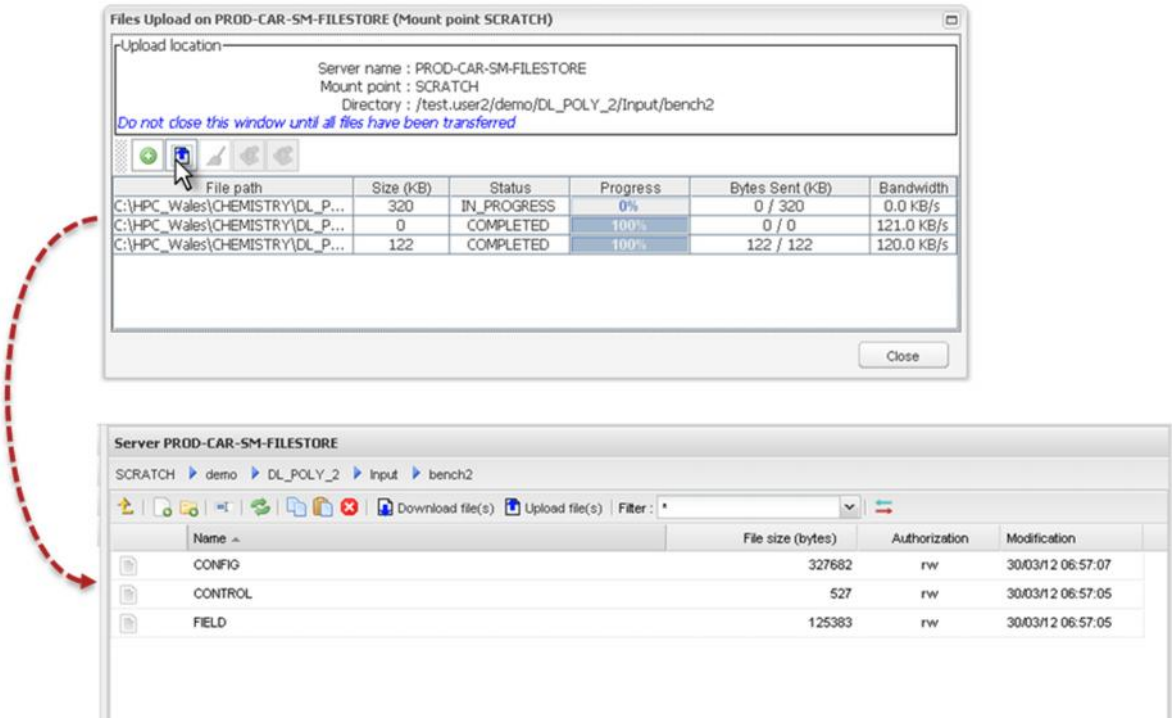


There are two steps within the upload window:

- Clicking on the selection button ( ) will open a further pop-up window. This runs on the desktop and is used to locate and select the files or folders to be transferred. The selection can be repeated to locate files from different locations that are to be transferred concurrently. Click **Open** when all files are chosen.



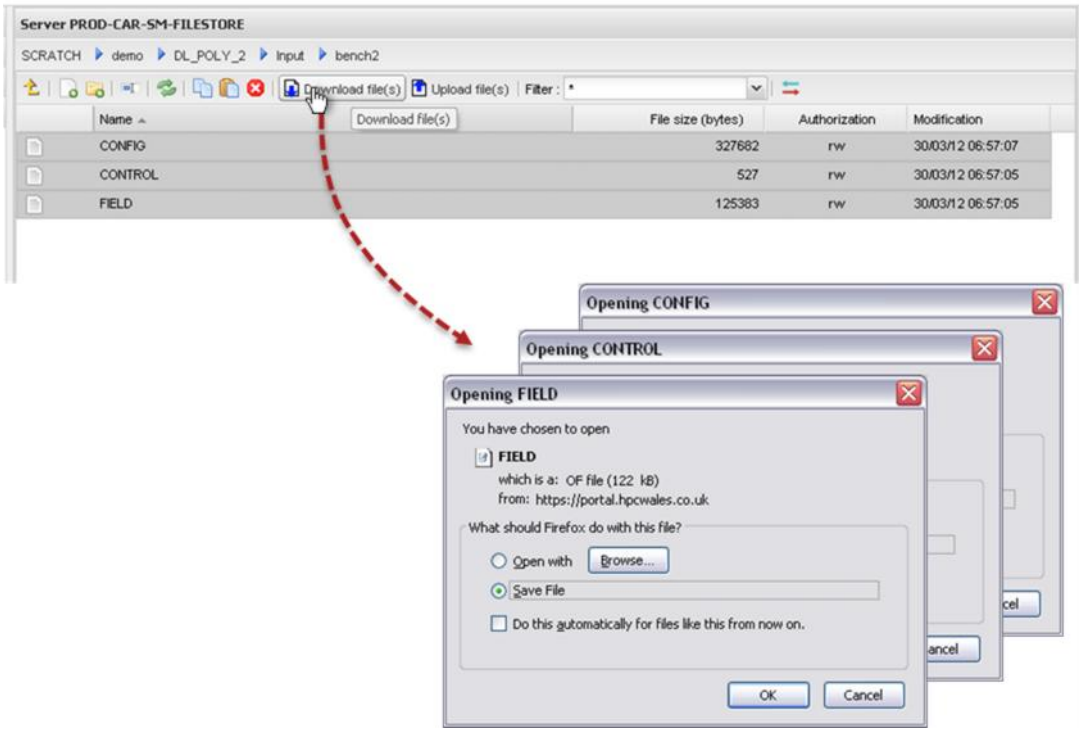
- Click on the Upload button to start the transfer. This action initiates a single direct transfer from the desktop machine to the target filesystem. There is no staging at the web server and files can be immediately referenced by workflow tasks running on any compute sub-system.



9.5.3 Downloading files

Navigate to the location where you want to select the files or folder to be downloaded to the desktop. Single and multiple files can be downloaded as a single action.

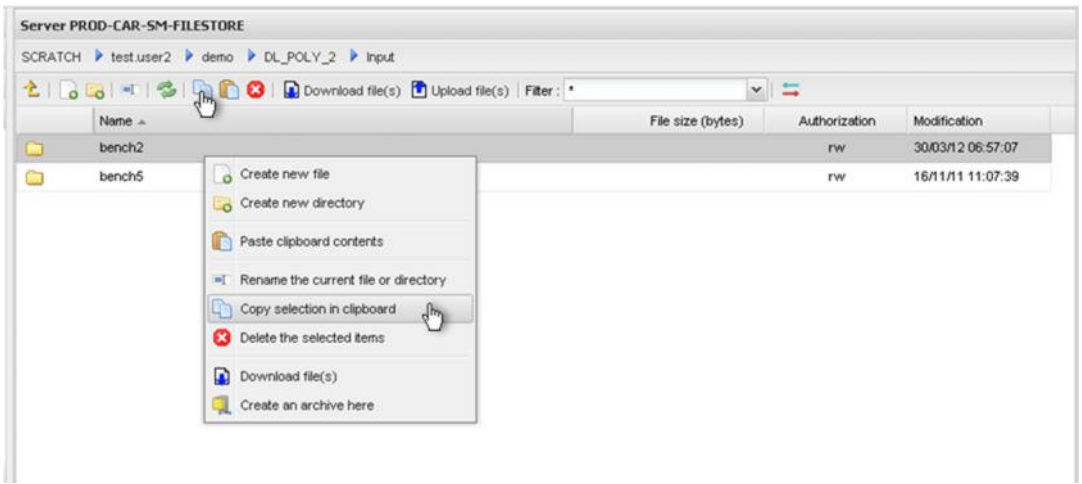
- Click on the file name to select for transfer. Choose multiple files by holding the shift (contiguous) or Ctrl key when selecting.
- Click on the Download file(s) button to initiate the transfer of the selected files. Each transfer will generate a pop-up on the desktop to treat the incoming file. The function of this window will depend on the browser type.



9.5.4 Copying files and directories

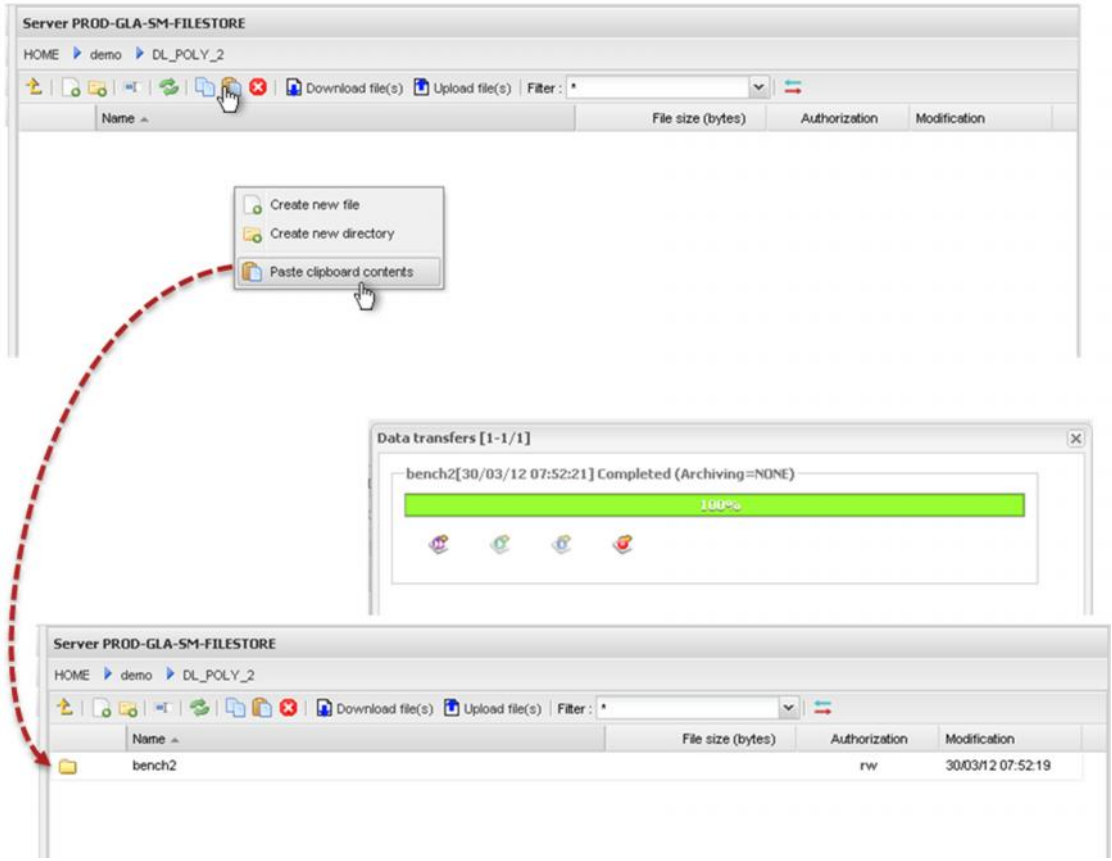
Files and directories can be copied and pasted between any SynfiniWay mount points across the entire HPC Wales system (where permitted). Select the items to transfer and choose **Copy selection in clipboard** from the drop-down list (right-click on name) or the associated button on the explorer toolbar.

- This example shows a directory being selected to copy from the users scratch area at the Cardiff site.



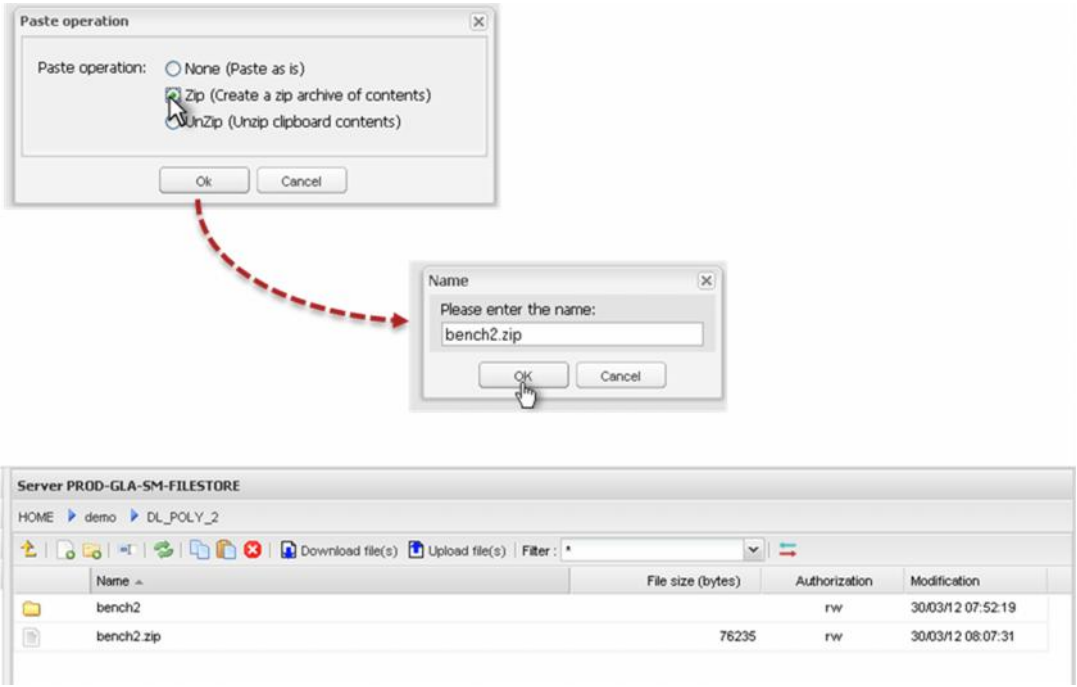
Next find the destination for the copied items.

- In the example below a location in the users home directory in Glamorgan is selected.
- A transfer progress bar is shown during the copy.
- The destination explorer view is refreshed when the transfer has completed.



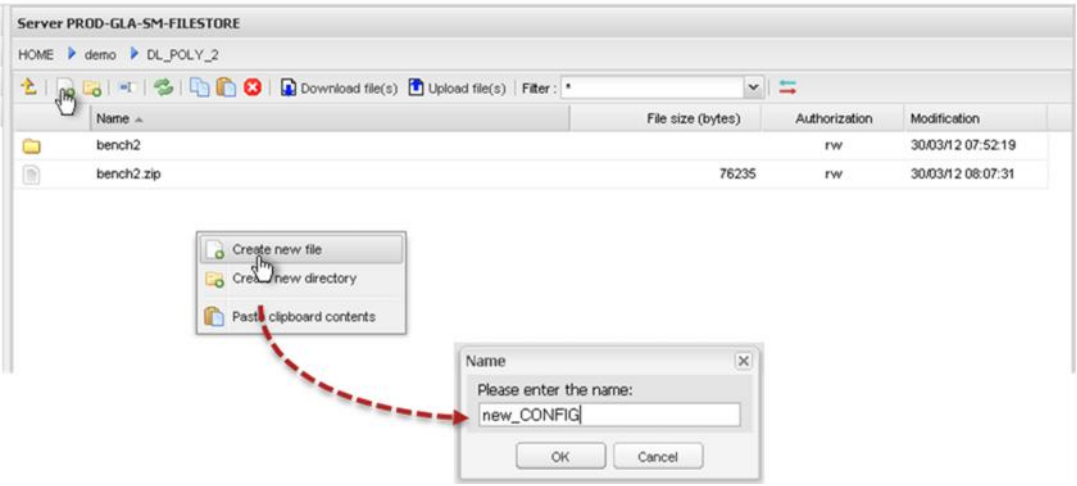
At the start of the copy you will be prompted with options for certain transformations applied on the paste operation. By default the transfer replicates the copied files and folders at the destination.

- Options are currently provided for compressing and uncompressing the contents of the clipboard.
- Equally this mechanism can be used you to pack files at the same location prior to download, or expand after upload, for instance.

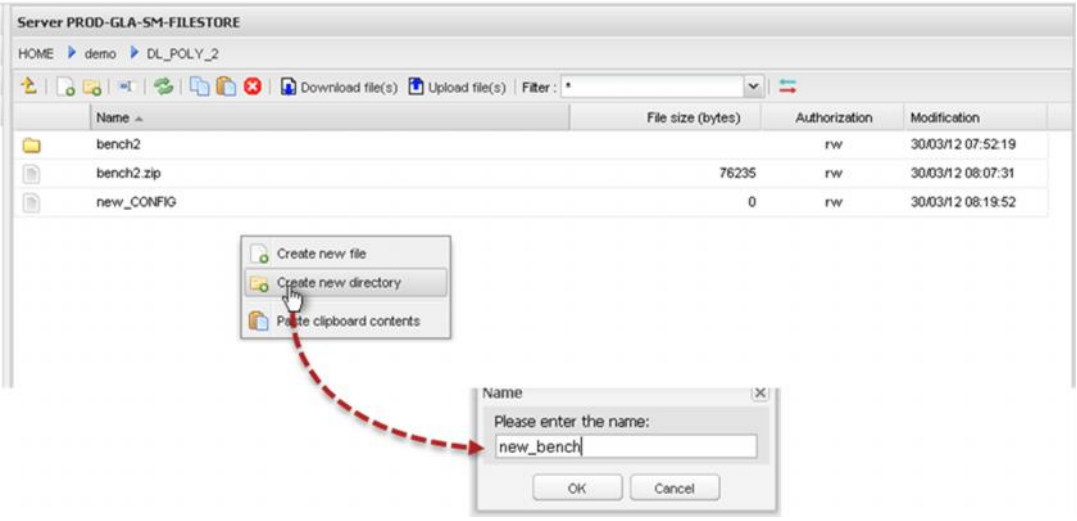


9.5.5 Creating and deleting files and directories

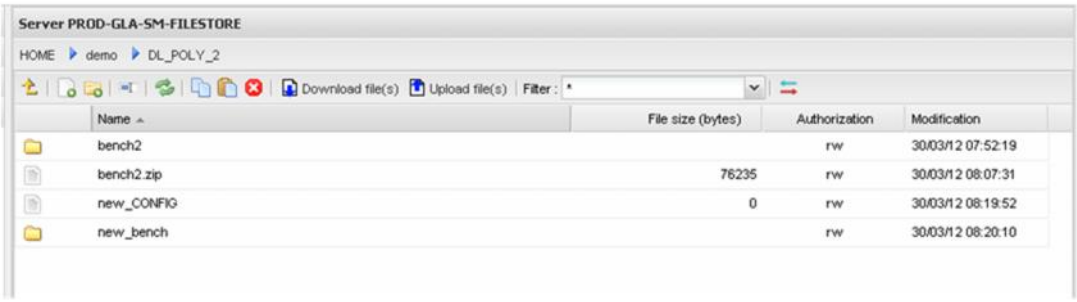
It is possible to open a new file on the remote filesystem using the **Create new file** function. This will create a file entry in the filesystem that can be edited using the in-built text editor (see section 9.5.6).



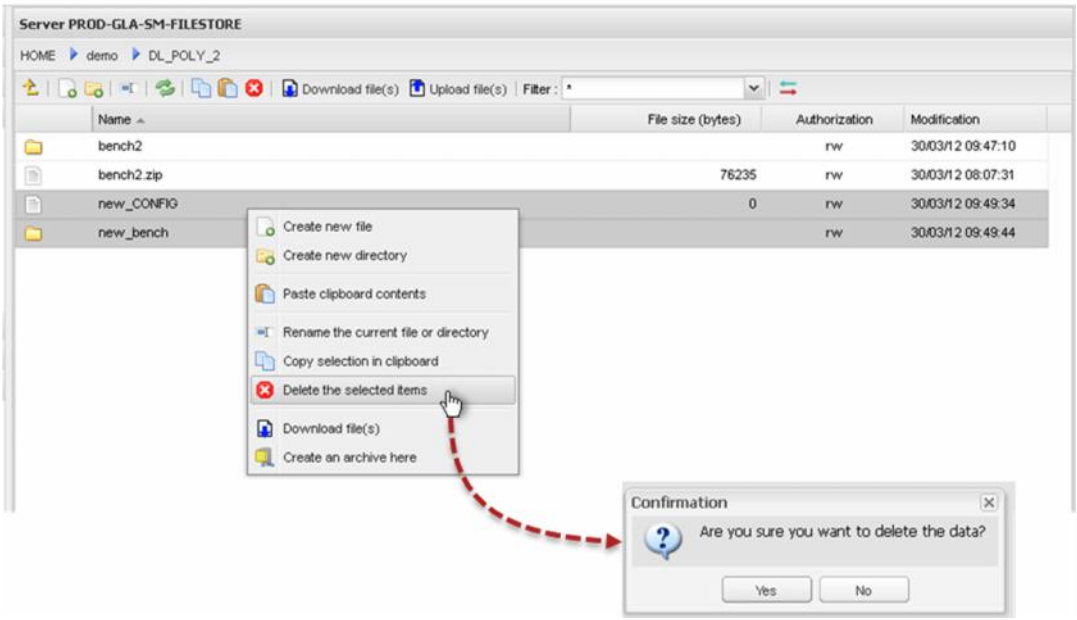
Similarly a directory or folder can be created on the remote filesystem, for instance to provide a new location for uploading files, set a destination for workflow output, or the place to pasting files from the clipboard.



After the file or directory is created the view on the remote filesystem is automatically refreshed.



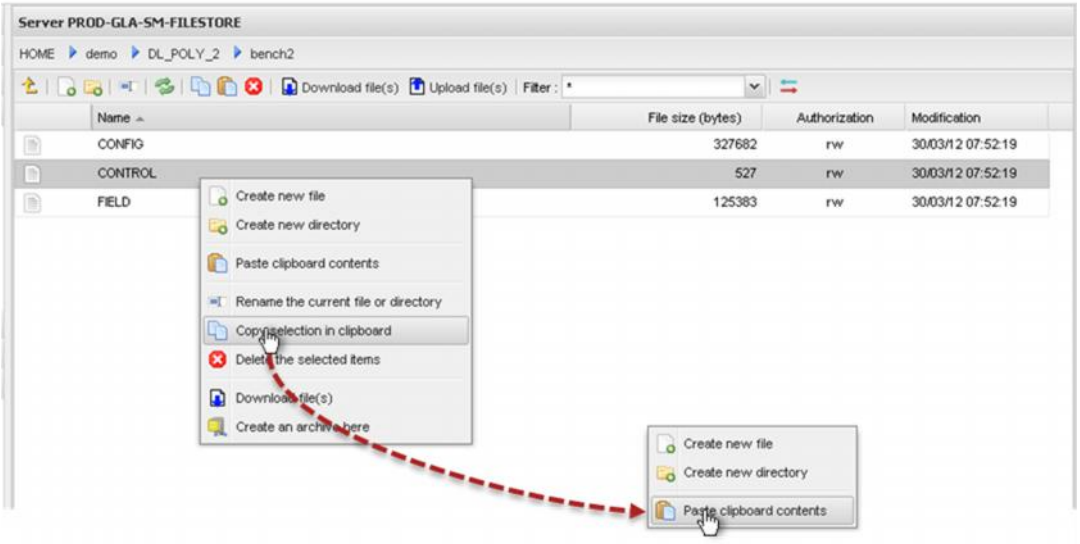
Deleting a file or directory is done by selecting the item/s and choosing the **Delete** button. A confirmation pop-up window will be displayed.



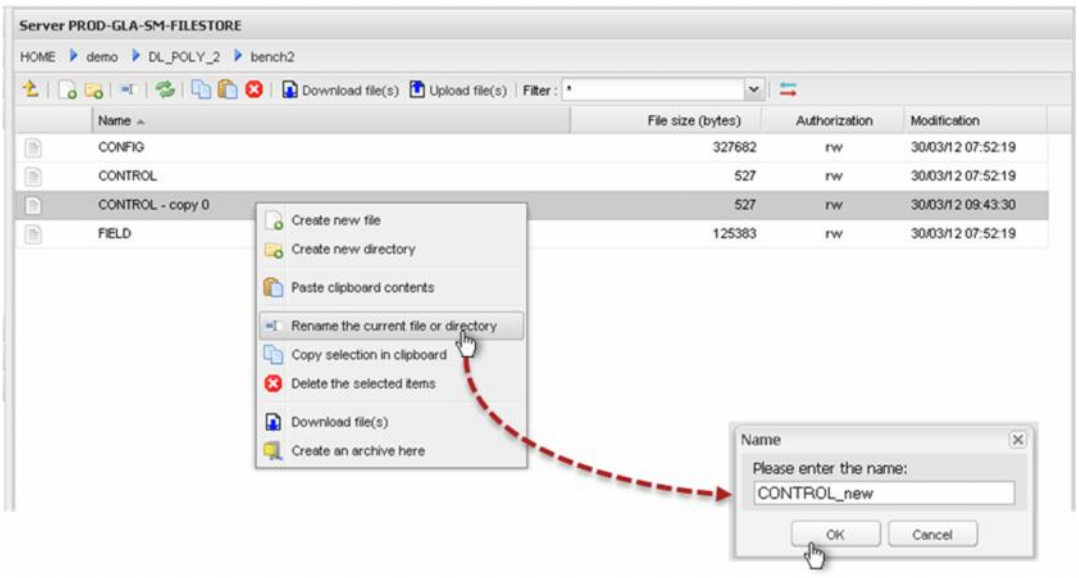
9.5.6 Editing files

The SynfiniWay web client provides a basic built-in text editor that allows you to create and modify remote files. To illustrate this capability the following worked example describes how an original input text file is modified for a subsequent test.

First, a copy if created of the starting task input file. A copy/paste in place automatically produces a new file with a name extension.



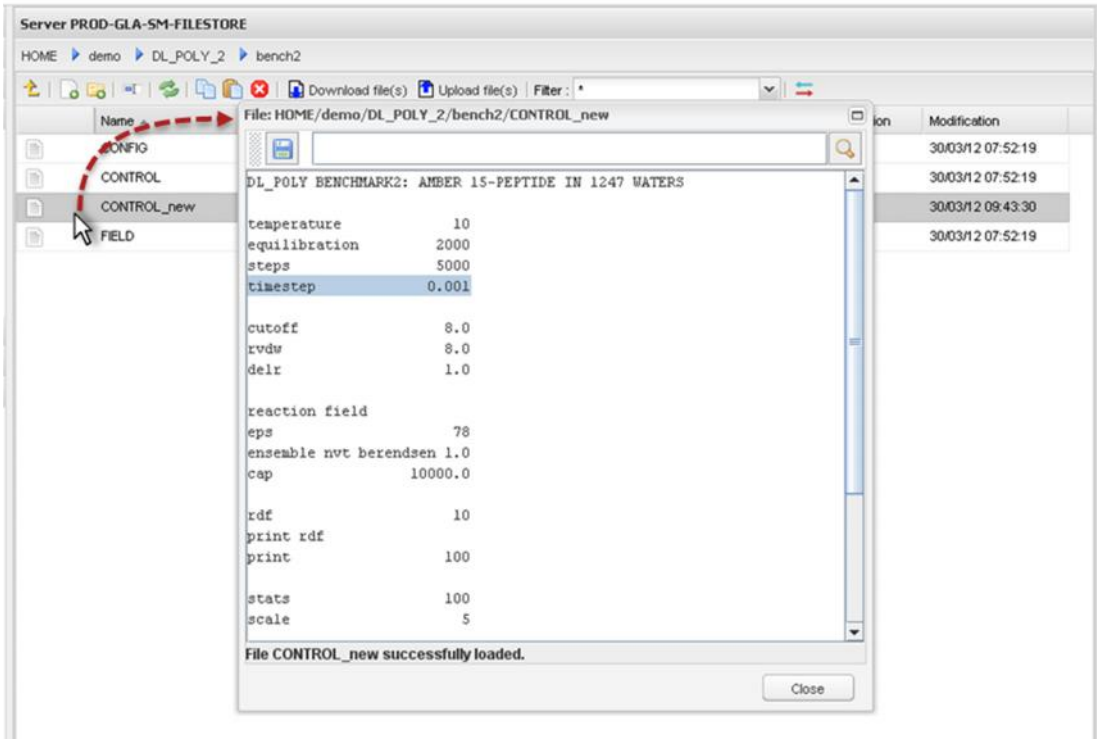
This file copy can then be renamed as needed.



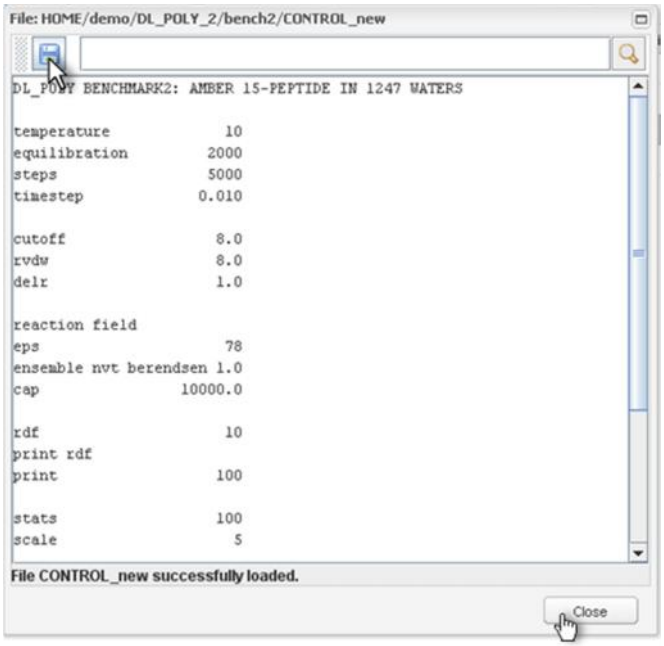
Double-click on the new file name to open the text editor.

- In this example we will change the timestep value as highlighted.

- The usual Ctrl-C and Ctrl-V controls can be used within this editor to copy and insert text.



When the changes are completed save the file by clicking the top-left icon.



Finally, close the editor by clicking on the **Close** button. This new file can now be referenced as a workflow input.

9.6 Developing Workflows

Use of the SynfiniWay capabilities for service and workflow development is described in the following product manuals:

- **SynfiniWay Workflow Editor's Guide, Version 4.0**
- **SynfiniWay Service Administration Guide, Version 4.0**

Copies of these manuals, and any supplemental training material, can be obtained from the HPC Wales system administration group.

The section may be expanded in future editions to describe the conventional *best-practice* approach to designing and developing SynfiniWay workflows. Subjects to be covered will include:

- Identifying the structure of your existing processes.
- Specifying the individual stages within the process as services.
- Creating and publishing services.
- Creating and publishing workflows.

10 Appendix I. HPC Wales Sites

SITE	DETAILS
CARDIFF HUB	**Login Nodes** login.hpcwales.co.uk log001.hpcwales.co.uk log002.hpcwales.co.uk log002.hpcwales.co.uk **Queue Names** q_cf_htc_work
SWANSEA HUB	*Dylan Thomas Centre(Swansea)* *Details of Login Nodes and Queue Names to be to be confirmed*
ABERYSTWYTH TIER-1	**Login Nodes** *(Aberystwyth University Campus or via login.hpcwales.co.uk)* ab-log-001 ab-log-002 **Queue Names** q_ab_mpc_all
BANGOR TIER-1	**Login Nodes** *(Bangor University Campus or via login.hpcwales.co.uk)* ba-log-001 ba-log-002 **Queue Names** q_ba_mpc_all
GLAMORGAN TIER-1	**Login Nodes** *(Glamorgan University Campus or via login.hpcwales.co.uk)* gl-log-001 gl-log-002 **Queue Names** q_gl_mpc_all

SWANSEA MET TIER-2A	**Login Nodes** *(via login.hpcwales.co.uk)* sm-log-001 sm-log-002 **Queue Names** q_sm_spc_all
NEWPORT TIER-2A	*To be confirmed*
GLYNDWR TIER-2A	**Login Nodes** *(via login.hpcwales.co.uk)* gd-log-001 gd-log-002 **Queue Names** q_gd_spc_all

11 Appendix II. Intel Compiler Flags

Code Optimisations	
Compiler FLAG	
	Unless specified, -O2 is assumed
-O0	Disables all optimizations
-O1	Enables optimizations for speed and disables some optimizations that increase code size and affect speed.
-O2	Enables optimizations for speed. This is the generally recommended optimization level. Vectorization is enabled at O2 and higher levels.
-O3	Performs O2 optimizations and enables more aggressive loop transformations such as Fusion, Block-Unroll-and-Jam, and collapsing IF statements.
-fast	Fast is a collection of compiler flags which, for the latest version of the compiler, expands to "`-O3 -xhost -ipo -static -no-prec-div`"
-no-prec-div	This option enables optimizations that give slightly less precise results than full IEEE division i.e. division is transformed into multiplication by the reciprocal of the denominator
	Processor Optimisations
	Unless specified, SSE level 2 is assumed
-xhost	Produces code optimised to run on the processor on which it was compiled
–xsse4.2	Produces code optimised to run on SSE level 4.2 capable processors
	Inter Procedural Optimisation
	Unless specified, IPO is disabled
-ipo	This option enables interprocedural optimization between files. When you specify this option, the compiler performs inline function expansion for calls to functions defined in separate files.
	Profile Guided Optimisation
	Unless specified, PGO is disabled
	PGO consists of three phases: Phase one is to build an instrumented executable; Phase two is to run that instrumented executable one or more times (on a range of typical data sets, or on a range of typical parameters) to generate one or more typical execution profiles; Phase three is to build an optimised executable based on information from the generated execution

	profiles.
-prof-gen	This option creates an instrumented executable (phase one)
-prof-use	This option creates an optimised executable (phase three)
	OTHER
-static	This option prevents linking with shared libraries, causing executables to link all libraries statically

Notes:

The Cardiff head nodes contain Intel(R) Xeon(R) CPU X5670 @ 2.93GHz processors which are SSE level 4.2 capable.

The Cardiff HTC compute nodes contain Intel(R) Xeon(R) CPU X5650 @ 2.67GHz processors which are SSE level 4.2 capable

SSE instructions allow the processor to execute the same instruction on multiple data elements at the same time (SIMD)

- SSE level 1 added 70 SIMD instructions to the processor architecture (from 1999 onwards)
- SSE level 2 added a further 144 SIMD instructions to the processor architecture (from 2001 onwards)
- SSE level 3 added a further 13 SIMD instructions to the processor architecture (from February 2004 onwards)
- SSSE level 3 added a further 16 instructions to the processor architecture (from June 2006 onwards)
- SSE level 4.1 added a further 47 instructions to the processor architecture (from July 2006 onwards)

SSE level 4.2 added a further 7 instructions to the processor architecture (from November 2008 onwards)

12 Appendix III. Common Linux Commands

The most common Linux commands are shown in the table below.

COMMAND	DESCRIPTION
cat [filename]	Display file's contents to the standard output device (usually your monitor).
cd /directorypath	Change to directory.
chmod [options] mode filename	Change a file's permissions.
chown [options] filename	Change who owns a file.
clear	Clear a command line screen/window for a fresh start.
cp [options] source destination	Copy files and directories.
date [options]	Display or set the system date and time.
df [options]	Display used and available disk space.
du [options]	Show how much space each file takes up.
file [options] filename	Determine what type of data is within a file.
find [pathname] [expression]	Search for files matching a provided pattern.
grep [options] pattern [filesname]	Search files or output for a particular pattern.
kill [options] pid	Stop a process. If the process refuses to stop, use kill -9 pid.
less [options] [filename]	View the contents of a file one page at a time.
ln [options] source [destination]	Create a shortcut.
locate filename	Search a copy of your filesystem for the specified filename.
lpr [options]	Send a print job.
ls [options]	List directory contents.
man [command]	Display the help information for the specified command.
mkdir [options] directory	Create a new directory.
mv [options] source destination	Rename or move file(s) or directories.
passwd [name [password]]	Change the password or allow (for the system administrator) to change any password.

ps [options]	Display a snapshot of the currently running processes.
pwd	Display the pathname for the current directory.
rm [options] directory	Remove (delete) file(s) and/or directories.
rmdir [options] directory	Delete empty directories.
ssh [options] user@machine	Remotely log in to another Linux machine, over the network. Leave an ssh session by typing **exit**.
su [options] [user [arguments]]	Switch to another user account.
tail [options] [filename]	Display the last *n* lines of a file (the default is 10).
tar [options] filename	Store and extract files from a tarfile (.tar) or tarball (.tar.gz or .tgz).
top	Displays the resources being used on your system. Press q to exit.
touch filename	Create an empty file with the specified name.
who [options]	Display who is logged on.

13 Appendix IV. HPC Wales Software Portfolio

After some 12 months of operation. HPC Wales has accumulated a wide variety of software available to its user community; we continually update the associated application packages, compilers, communications libraries, tools, and math libraries. To facilitate this task and to provide a uniform mechanism for accessing different revisions of software, HPC Wales uses the modules utility (see section 5.4). This appendix lists the software available via the module system as of 28th January 2013. This software is classified below under the headings of compilers, languages, libraries, tools, benchmarks and applications. The latter are further broken down into the sector areas of chemistry, creative (industries), Financial, Genomics (life sciences), Materials and Environment.

13.1 Compilers

Compiler	Description		
compiler	A compiler is a computer program (or set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language, often having a binary form known as object code).		
	Intel	intel-11.1.072	intel-11.1
	intel-12.0.084	intel-12.0	gnu
	gnu-4.1.2	gnu-4.6.2	portland-12.5

13.2 Languages

Language	Description		
perl	Perl is a high-level, general-purpose, interpreted, dynamic programming language.		
	5.14.2		
python	Python is a general-purpose, high-level programming language whose design philosophy emphasizes code readability.		
	2.6.7	2.7.3	2.7.3-gnu-4.6.2
R	R is an open source programming language and software environment for statistical computing and graphics. The R language is widely used among statisticians for developing statistical software and data analysis.		
	2.13.1	2.14.2	2.14.1
Java	Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java is as of 2012 one of the most popular programming languages in		

	use. Not to be confused with JavaScript.
	`1.6.0_31`

13.3 Libraries

Library	Description
`atlas`	
	`3.10.0-gnu-4.6.2(default)`
`beagle`	BEAGLE is a high-performance library that can perform the core calculations at the heart of most Bayesian and Maximum Likelihood phylogenetics packages. It can make use of highly-parallel processors such as those in graphics cards (GPUs) found in many PCs. The project involves an open API and fast implementations of a library for evaluating phylogenetic likelihoods (continuous time Markov processes) of biomolecular sequence evolution. The aim is to provide high performance evaluation 'services' to a wide range of phylogenetic software, both Bayesian samplers and Maximum Likelihood optimizers. This allows these packages to make use of implementations that make use of optimized hardware such as graphics processing units.
	`1075(default)`
`beamnrc`	BEAMnrc is a Monte Carlo simulation system (Med. Phys. 22,1995,503 – 524) for modelling radiotherapy sources which was developed as part of the OMEGA project to develop 3-D treatment planning for radiotherapy (with the University of Wisconsin). BEAMnrc is built on the EGSnrc Code System.
	`4-2.3.2`
`boost`	Boost provides free peer-reviewed portable C++ source libraries.
	`boost_1_51_0`
`cairo`	Cairo is a 2D graphics library with support for multiple output devices. Currently supported output targets include the X Window System (via both Xlib and XCB), Quartz, Win32, image buffers, PostScript, PDF, and SVG file output.
	`1.12.0(default)`
`chroma`	The Chroma package supports data-parallel programming constructs for lattice field theory and in particular lattice QCD. It uses the SciDAC QDP++ data-parallel programming (in C++) that presents a single high-level code image to the user, but can generate highly optimized code for many architectural systems.
	`3.38.0`
`delft3d`	Delft3D is a flexible integrated modelling suite, which simulates two-

	dimensional (in either the horizontal or a vertical plane) and three-dimensional flow, sediment transport and morphology, waves, water quality and ecology and is capable of handling the interactions between these processes. The suite is designed for use by domain experts and non-experts alike, which may range from consultants and engineers or contractors, to regulators and government officials, all of whom are active in one or more of the stages of the design, implementation and management cycle.
	`1301-gnu-4.6.2(default)  5.00.10-gnu-4.6.2`
egsnrc	This module contains classes that model particle sources.
	`4-2.3.2`
expat	Expat is an XML parser library written in C. It is a stream-oriented parser in which an application registers handlers for things the parser might find in the XML document (like start tags).
	`2.0.1(default)`
fftw	FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, i.e. the discrete cosine/sine transforms or DCT/DST).
	`3.3(default)  3.3-serial`
fontconfig	Fontconfig is a library for configuring and customizing font access.
	`2.8.0(default)`
freetype	`FreeType 2` is a software font engine that is designed to be small, efficient, highly customizable, and portable while capable of producing high-quality output (glyph images). It can be used in graphics libraries, display servers, font conversion tools, text image generation tools, and many other products as well.
	`2.4.9(default)  2.4.9-gnu`
ga	The Global Arrays (GA) toolkit provides an efficient and portable "shared-memory" programming interface for distributed-memory computers. Each process in a MIMD parallel program can asynchronously access logical blocks of physically distributed dense multi-dimensional arrays, without need for explicit cooperation by other processes. Unlike other shared-memory environments, the GA model exposes to the programmer the non-uniform memory access (NUMA) characteristics of HPC systems and acknowledges that access to a remote portion of the shared data is slower than to the local portion. Global Arrays have been designed to complement rather than substitute for the message-passing programming model. The programmer is free to use both the shared-memory and message-passing paradigms in the same

	program, and to take advantage of existing message-passing software libraries. Global Arrays are compatible with the Message Passing Interface (MPI). The Global Arrays toolkit has been in the public domain since 1994. It has been actively supported and employed in several large codes since then e.g. NWChem, GAMESS-UK, Molpro etc.
	`5.0.2(default)`
GDAL	GDAL - Geospatial Data Abstraction Library - is a translator library for raster geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. As a library, it presents a single abstract data model to the calling application for all supported formats.
	`1.9.1` `1.9.1-gnu-4.6.2`
GEOS	GEOS (Geometry Engine - Open Source) is a C++ port of the Java Topology Suite (JTS). As such, it aims to contain the complete functionality of JTS in C++. This includes all the OpenGIS Simple Features for SQL spatial predicate functions and spatial operators, as well as specific JTS enhanced topology functions.
	`3.3.5` `3.3.5-gnu-4.6.2`
geant	`Geant4` is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science.
	`4.9.5`
gmp	GMP is a free library for arbitrary precision arithmetic, operating on signed integers, rational numbers, and floating point numbers. There is no practical limit to the precision except the ones implied by the available memory in the machine GMP runs on. GMP has a rich set of functions, and the functions have a regular interface. The main target applications for GMP are cryptography applications and research, Internet security applications, algebra systems, computational algebra research, etc.
	`5.0.2`
google-sparsehash	An extremely memory-efficient hash_map implementation (2 bits/entry overhead)! The `SparseHash` library contains several `hash-map` implementations, including implementations that optimize for pace or speed. These hashtable implementations are similar in API to SGI's `hash_map` class and the tr1 `unordered_map` class, but with different performance characteristics. It's easy to replace `hash_map` or `unordered_map` by `sparse_hash_map` or `dense_hash_map` in C++ code.
	`sparsehash-2.0.1/2.0.1`

gsl	
	1.15(default)
gts	GNU Triangulated Surface Library. It is an Open Source Free Software Library intended to provide a set of useful functions to deal with 3D surfaces meshed with interconnected triangles.
	120706(default)
hdf5	HDF5 is a data model, library, and file format for storing and managing data. It supports an unlimited variety of datatypes, and is designed for flexible and efficient I/O and for high volume and complex data. HDF5 is portable and is extensible, allowing applications to evolve in their use of HDF5. The HDF5 Technology suite includes tools and applications for managing, manipulating, viewing, and analyzing data in the HDF5 format.
	1.8.6(default) 1.8.6-serial 1.8.6-shared 1.8.8-c++ 1.8.8-shared 1.8.9-shared 1.8.9-shared-gnu-4.6.2
jasper	The JasPer Project is an open-source initiative to provide a free software-based reference implementation of the codec specified in the JPEG-2000 Part-1 standard (i.e., ISO/IEC 15444-1). This project was started as a collaborative effort between Image Power,nc. and the University of British Columbia. Presently, the on-going maintenance and development of the JasPer software is being co-ordinated by its principal author, Michael Adams, who is affiliated with the Digital Signal Processing Group (DSPG) in the Department of Electrical and Computer Engineering at the University of Victoria. JasPer includes a software-based implementation of the codec specified in the JPEG-2000 Part-1 standard (i.e., ISO/IEC 15444-1). The JasPer software is written in the C programming language.
	1.900.1(default)
lapack	LAPACK (Linear Algebra PACKage) is a software library for numerical linear algebra. It provides routines for solving systems of linear equations and linear least squares, eigenvalue problems, and singular value decomposition. It also includes routines to implement the associated matrix factorizations such as LU, QR, Cholesky and Schur decomposition. LAPACK was originally written in FORTRAN 77, but moved to Fortran 90 in version 3.2 (2008). The routines handle both real and complex matrices in both single and double precision.
	3.4.2-gnu-4.6.2(default)
libffi	The libffi library provides a portable, high level programming interface to various calling conventions. This allows a programmer to call any function specified by a call interface description at run-time. FFI stands for Foreign Function Interface. A foreign function interface is the popular name for the interface that allows code written in one language to

	call code written in another language. The `libffi` library really only provides the lowest, machine dependent layer of a fully featured foreign function interface. A layer must exist above `libffi` that handles type conversions for values passed between the two languages.
	`3.0.11-gnu-4.6.2`
libgeotiff	GeoTIFF represents an effort by over 160 different remote sensing, GIS, cartographic, and surveying related companies and organizations to establish a TIFF based interchange format for georeferenced raster imagery.
	`1.4.0`
libpng	`libpng` is the official PNG reference library. It supports almost all PNG features, is extensible, and has been extensively tested for over 16 years.
	`1.2.50 1.5.10(default)`
libtool	GNU libtool is a generic library support script. Libtool hides the complexity of using shared libraries behind a consistent, portable interface.
	`2.4.2-gnu-4.1.2(default)`
libunwind	A portable and efficient C programming interface (API) to determine the call-chain of a program. The API additionally provides the means to manipulate the preserved (callee-saved) state of each call-frame and to resume execution at any point in the call-chain (non-local goto). The API supports both local (same-process) and remote (across-process) operation.
	`1.0.1`
libxml2	Libxml2 is the XML C parser and toolkit developed for the Gnome project (but usable outside of the Gnome platform), it is free software available under the MIT License. XML itself is a metalanguage to design markup languages, i.e. text language where semantic and structure are added to the content using extra "markup" information enclosed between angle brackets. HTML is the most well-known markup language. Though the library is written in C a variety of language bindings make it available in other environments.
	`2.6.19`
mcr	MATLAB Compiler Runtime (MCR) provides support for MATLAB applications as an executable or a shared library. Executables and libraries are created with the MATLAB compiler use the MCR runtime engine.
	`v717(default)`
mpc	Gnu `Mpc` is a C library for the arithmetic of complex numbers with arbitrarily high precision and correct rounding of the result. It extends the principles of the IEEE-754 standard for fixed precision real floating point numbers to complex numbers, providing well-defined semantics for every operation. At the same time, speed of operation at high precision is a major design goal.

	0.9
mpfr	The MPFR library is a C library for multiple-precision floating-point computations with correct rounding. MPFR is based on the GMP multiple-precision library. The main goal of MPFR is to provide a library for multiple-precision floating-point computation which is both efficient and has a well-defined semantics. It copies the good ideas from the ANSI/IEEE-754 standard for double-precision floating-point arithmetic (53-bit significant).
	3.1.0
mpi	Message Passing Interface (MPI) is a standardized and portable message-passing system designed by a group of researchers from academia and industry to function on a wide variety of parallel computers. The standard defines the syntax and semantics of a core of library routines useful to a wide range of users writing portable message-passing programs in Fortran 77 or the C programming language. Several well-tested and efficient implementations of MPI include some that are free and in the public domain.
	intel(default) intel-4.0 intel-4.0.3.008 intel-4.1 intel-4.1.0.030 mpich2-1.5-gnu-4.6.2 mpich2-1.5.1-gnu-4.1.2 mpich2-1.5.1-gnu-4.6.2 mvapich1 mvapich2 openmpi openmpi-1.4.2 openmpi-1.5.4 platform platform-8.1
mpi4py	MPI for Python provides bindings of the Message Passing Interface (MPI) standard for the Python programming language, allowing any Python program to exploit multiple processors. The package is constructed on top of the MPI-1/2 specifications and provides an object oriented interface which closely follows MPI-2 C++ bindings. It supports point-to-point (sends, receives) and collective (broadcasts, scatters, gathers) communications of a pickable Python object, as well as optimized communications of Python object exposing the single segment buffer interface (NumPy arrays, builtin bytes/string/array objects).
	1.3
mpiP	mpiP is a lightweight profiling library for MPI applications. Because it only collects statistical information about MPI functions, mpiP generates considerably less overhead and much less data than tracing tools. All the information captured by mpiP is task-local. It only uses communication during report generation, typically at the end of the experiment, to merge results from all of the tasks into one output file.
	3.3
muparser	muParser is an extensible high performance math expression parser library written in C++. It works by transforming a mathematical expression into bytecode and pre-calculating constant parts of the expression.
	2.2.2

ncl	The Nexus Class Library (NCL) is a C++ library for interpreting data files created according to the NEXUS file format used in phylogenetic systematics and molecular evolution.
	`2.1.17(default)`
netCDF	NetCDF is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.
	`3.6.3(default) 4.1.3 4.1.3-shared 4.1.3-shared-gnu-4.6.2`
nose	Nose is a nicer testing for python. Nose extends unittest to make testing easier.
	`1.2.1-gnu-4.6.2(default)`
numpy	NumPy is the fundamental package for scientific computing with Python.
	`1.6.2-gnu-4.6.2(default)`
octave	GNU Octave is a high-level interpreted language, primarily intended for numerical computations. It provides capabilities for the numerical solution of linear and nonlinear problems, and for performing other numerical experiments. It also provides extensive graphics capabilities for data visualization and manipulation. Octave is normally used through its interactive command line interface, but it can also be used to write non-interactive programs. The Octave language is quite similar to Matlab so that most programs are easily portable.
	`3.6.3-gnu-4.6.2(default)`
openssl	OpenSSL is a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library.
	`1.0.0g`
paraFEM	A portable library for parallel finite element analysis.
	`2.0.819(default)`
parallel-netCDF	Parallel netCDF (PnetCDF) is a library providing high-performance I/O while still maintaining file-format compatibility with Unidata's NetCDF.
	`1.2.0(default)`

pcre	The PCRE library is a set of functions that implement regular expression pattern matching using the same syntax and semantics as Perl 5. PCRE has its own native API, as well as a set of wrapper functions that correspond to the POSIX regular expression API.
	`8.32-gnu-4.6.2(default)`
pdt	Program Database Toolkit (PDT) is a framework for analyzing source code written in several programming languages and for making rich program knowledge accessible to developers of static and dynamic analysis tools. PDT implements a standard program representation, the program database (PDB), that can be accessed in a uniform way through a class library supporting common PDB operations.
	`3.17(default)`
petsc	PETSc is a suite of data structures and routines for the scalable (parallel) solution of scientific applications modelled by partial differential equations. It supports MPI, shared memory pthreads, and NVIDIA GPUs, as well as hybrid MPI-shared memory pthreads or MPI-GPU parallelism.
	`3.1 3.2(default)`
pixman	Pixman is the pixel-manipulation library for X and cairo.
	`0.26.2(default)`
proj	PROJ.4 Cartographic Projections library.
	`4.8.0`
pycogent	PyCogent is a software library for genomic biology. It is a fully integrated and thoroughly tested framework for: controlling third-party applications; devising workflows; querying databases; conducting novel probabilistic analyses of biological sequence evolution and generating publication quality graphics. It is distinguished by many unique built-in capabilities (such as true codon alignment) and the frequent addition of entirely new methods for the analysis of genomic data.
	`1.5.1(default)`
qdp++	The QDP++ interface provides a data-parallel programming environment suitable for Lattice QCD where operations in operator/infix form can be applied on lattice wide objects. The interface provides a level of abstraction such that high-level user code written above the interface can be run unchanged on a single processor workstation or a collection of multiprocessor nodes with parallel communications. Architectural dependencies are hidden below the interface. A variety of types for the site elements are provided. To achieve good performance, overlapping communication and computation primitives are provided.

	1.36.1
qmp	The QMP project is a national effort to provide a high performance message passing interface on various hardware platforms for Lattice QCD computing. This message passing interface aims to provide channel oriented communication end points to communication readers and writers with low latency and high bandwidth. QMP is tailored to the repetitive and predominantly nearest neighbour communication patterns of lattice QCD calculations.
	2.1.6
Rmpi	Rmpi provides an interface (wrapper) to MPI APIs. It also provides interactive R slave environment.
	0.6-1
scipy	SciPy (pronounced "Sigh Pie") is open-source software for mathematics, science, and engineering. It is also the name of a very popular conference on scientific programming with Python. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The SciPy library is built to work with NumPy arrays, and provides many user-friendly and efficient numerical routines such as routines for numerical integration and optimization.
	0.11.0-gnu-4.6.2(default)
xerces	A processor for parsing, validating, serializing and manipulating XML, written in C++
	3.1.1
yasm	The Yasm Modular Assembler Project. Yasm is a complete rewrite of the NASM assembler under the “new” BSD License. Yasm currently supports the x86 and AMD64 instruction sets, accepts NASM and GAS assembler syntaxes, outputs binary, ELF32, ELF64, 32 and 64-bit Mach-O, RDOFF2, COFF, Win32, and Win64 object formats, and generates source debugging information in STABS, DWARF 2, and CodeView 8 formats.
	1.2.0(default)
zlib	A Massively Spiffy Yet Delicately Unobtrusive Compression Library
	1.2.7

13.4 Tools

Tool	Description

allinea-ddt	Allinea DDT is an advanced debugging tool available for scalar, multi-threaded and large-scale parallel applications.
antlr	ANTLR, ANother Tool for Language Recognition, is a language tool that provides a framework for constructing recognizers, interpreters, compilers, and translators from grammatical descriptions containing actions in a variety of target languages. ANTLR provides excellent support for tree construction, tree walking, translation, error recovery, and error reporting.
	2.7.7
autoconf	Autoconf is an extensible package of M4 macros that produce shell scripts to automatically configure software source code packages. These scripts can adapt the packages to many kinds of UNIX-like systems without manual user intervention. Autoconf creates a configuration script for a package from a template file that lists the operating system features that the package can use, in the form of M4 macro calls. Producing configuration scripts using Autoconf requires GNU M4. The configuration scripts produced by Autoconf are self-contained, so their users do not need to have Autoconf (or GNU M4).
	2.68
automake	Automake is a tool for automatically generating 'Makefile.in' files compliant with the GNU Coding Standards. Automake requires the use of autoconf.
	1.11.3
cmake	CMake is a family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces that can be used in the compiler environment of your choice.
	2.8.7
ferret	Ferret is the GNU data modeller. Ferret (formerly known as GerWin) is the Free Entity Relationship and Reverse Engineering Tool. It lets you create data models and implement them in a relational database. You can draw your data model via an entity-relation diagram, generate the tables from it (another graphical diagram), and then generate the SQL that creates such relational tables. Several SQL dialects are supported (for most popular free software database systems) and it is very easy to patch the sources to support more.
	6.72

ffmpeg	FFmpeg is a complete, cross-platform solution to record, convert and stream audio and video. It includes libavcodec - the leading audio/video codec library.
	`1.0(default)`
git	`Git` is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. `Git` is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.
	`1.7.2`
gnuplot	Gnuplot is a portable command-line driven graphing utility. It was originally created to allow scientists and students to visualize mathematical functions and data interactively, but has grown to support many non-interactive uses such as web scripting. It is also used as a plotting engine by third-party applications like Octave. Gnuplot has been supported and under active development since 1986.
	`4.6.0`
impi_collective_tune	Intel – The Interoperable Message Passing Interface.
intel-inspector-xe	Used for detecting memory and threading defects early in the development cycle.
ipm	IPM is a portable profiling infrastructure for parallel codes. It provides a low-overhead performance profile of the performance aspects and resource utilization in a parallel program. Communication, computation, and IO are the primary focus. While the design scope targets production computing in HPC centres, IPM has found use in application development, performance debugging and parallel computing education. The level of detail is selectable at runtime and presented through a variety of text and web reports.
	`0.983(default)`
mdtest	mdtest is an MPI-coordinated metadata benchmark test that performs open/stat/close operations on files and directories and then reports the performance.
	`1.8.3(default)`

mencoder	MEncoder is a free command line video decoding, encoding and filtering tool released under the GNU General Public License. It is a sibling of MPlayer, and can convert all the formats that MPlayer understands into a variety of compressed and uncompressed formats using different codecs.
	1.1
moa	MOA is a framework for learning from a continuous supply of examples, a data stream. Includes classification and clustering methods. Related to the WEKA project, also written in Java, while scaling to more demanding problems.
	20120301(default)
mpitest	
nco	The netCDF Operators (NCO) comprise a dozen standalone, command-line programs that take netCDF files as input, then operate (e.g., derive new data, average, print, hyperslab, manipulate metadata) and output the results to screen or files in text, binary, or netCDF formats. NCO aids manipulation and analysis of gridded scientific data. The shell-command style of NCO allows users to manipulate and analyse files interactively, or with simple scripts that avoid some overhead (and power) of higher level programming environments.
	4.1.0
ncview	Ncview is a visual browser for netCDF format files.
	2.1.1
ne	ne is a text editor based on the POSIX standard. ne is easy to use for the beginner, but powerful and fully configurable for the wizard, and most sparing in its resource usage.
	2.4
openss	OpenSpeedShop python interface package. This package is essentially another UI for OpenSpeedShop. Currently this is done by converting the APIs objects into OpenSpeedShop CLI (Command Line Interface) syntax passing them to a common parser and semantic handler. The rational for this method is to reduce duplicate code and make behaviour consistent across UIs.
	2.0.1(default)
paraview	ParaView is an open-source, multi-platform data analysis and visualization application. ParaView can quickly build visualizations

	to analyse data using qualitative and quantitative techniques. The data exploration can be done interactively in 3D or programmatically using ParaView's batch processing capabilities.
	`3.14.0`
`physica`	The PHYSICA toolkit is a modular suite of software components for the simulation of coupled physical phenomena in three spatial dimensions and time. Once installed, the program may be used for the prediction of continuum physical processes including both Computational Fluid Dynamics (CFD) and Computational Solid Mechanics (CSM), and interactions between the two. In release 2.10, PHYSICA contains modules to solve problems involving steady-state or transient fluid flow, heat transfer, phase-change and thermal stress. Structured or unstructured meshes may be used.
	`2.11(default)`
`ploticus`	A non-interactive software package for producing plots, charts, and graphics from data. `Ploticus` is good for automated or just-in-time graph generation, handles date and time data nicely, and has basic statistical capabilities. It allows significant user control over colours, styles, options and details.
	`2.41(default)`
`plotutils`	The GNU `plotutils` package contains software for both programmers and technical users. Its centrepiece is `libplot`, a powerful C/C++ function library for exporting 2-D vector graphics in many file formats, both vector and bitmap. On the X Window System, it can also do 2-D vector graphics animations. `libplot` is device-independent, in the sense that its API (application programming interface) does not depend on the type of graphics file to be exported. A Postscript-like API is used both for file export and for graphics animations. A `libplot` programmer needs to learn only one API: not the details of many graphics file formats.
	`2.6`
`pvm3`	PVM (Parallel Virtual Machine) is a software system that enables a collection of heterogeneous computers to be used as a coherent and flexible concurrent computational resource. The individual computers may be shared- or local-memory multiprocessors, vector supercomputers, specialized graphics engines, or scalar workstations, that may be interconnected by a variety of networks, such as ethernet, FDDI, etc. PVM support software executes on each machine in a user-configurable pool, and presents a unified, general, and powerful computational environment of concurrent applications. User programs written in C or Fortran are provided access to PVM through the use of calls to PVM library routines for functions such as process initiation, message transmission and reception, and

	synchronization via barriers or rendezvous. Users may optionally control the execution location of specific application components. The PVM system transparently handles message routing, data conversion for incompatible architectures, and other tasks that are necessary for operation in a heterogeneous, network environment. PVM is particularly effective for heterogeneous applications that exploit specific strengths of individual machines on a network. As a loosely coupled concurrent supercomputer environment, PVM is a viable scientific computing platform. The PVM system has been used for applications such as molecular dynamics simulations, superconductivity studies, distributed fractal computations, matrix algorithms, and in the classroom as the basis for teaching concurrent computing. It is true to say however that PVM has almost exclusively been superseded by OpenMP and MPI in all major parallel programming projects.
	`3.4.6`
`scalasca`	`Scalasca` is a software tool that supports the performance optimization of parallel programs by measuring and analysing their runtime behaviour. The analysis identifies potential performance bottlenecks – in particular those concerning communication and synchronization – and offers guidance in exploring their causes.
	`1.4(default)`
`sqlite`	SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world.
	`3071201(default)`
`subversion`	Subversion is an open source version control system.
	`1.7.2`
`tau`	TAU is a program and performance analysis tool framework being developed for the DOE Office of Science, ASC initiatives at LLNL, the ZeptoOS project at ANL, and the Los Alamos National Laboratory. TAU provides a suite of static and dynamic tools that provide graphical user interaction and inter-operation to form an integrated analysis environment for parallel Fortran, C++, C, Java, and Python applications. In particular, a robust performance profiling facility available in TAU has been applied extensively in the ACTS toolkit. Also, recent advancements in TAU's code analysis capabilities have allowed new static tools to be developed, such as an automatic instrumentation tool.
	`2.21`

texinfo	Texinfo is the official documentation format of the GNU project. It was invented by Richard Stallman and Bob Chassell many years ago, loosely based on Brian Reid's Scribe and other formatting languages of the time. It is used by many non-GNU projects as well. Texinfo uses a single source file to produce output in a number of formats, both online and printed (dvi, html, info, pdf, xml, etc.). This means that instead of writing different documents for online information and another for a printed manual, you need write only one document. And when the work is revised, you need revise only that one document. The Texinfo system is well-integrated with GNU Emacs.
	`4.13`
weka	Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.
	`3.6.6(default)`

13.5 Applications

The latter are further broken down into the sector areas of chemistry, creative (industries), Financial, Genomics (life sciences), Materials and Environment; note that a number of these areas currently contain no entries and are omitted below e.g. Financial, Materials. Many materials codes are currently classified under the "Chemistry" modules.

13.6 Chemistry

Code	Description
crystal	The CRYSTAL package performs *ab initio* calculations of the ground state energy, energy gradient, electronic wave function and properties of periodic systems. Hartree-Fock or Kohn-Sham Hamiltonians (that adopt an Exchange-Correlation potential following the postulates of Density-Functional theory) can be used. Systems periodic in 0 (molecules, 0D), 1 (polymers,1D), 2 (slabs, 2D), and 3 dimensions (crystals, 3D) are treated on an equal footing. In each case the fundamental approximation made is the expansion of the single particle wave functions ('Crystalline Orbital', CO) as a linear combination of Bloch functions (BF) defined in terms of local functions (hereafter indicated as 'Atomic Orbitals', AOs).
	`9.10`
dlpoly	DL_POLY is a general purpose serial and parallel molecular dynamics simulation package originally developed at Daresbury Laboratory by W. Smith and T.R. Forester under the auspices of the Engineering and Physical Sciences Research Council (EPSRC) for the EPSRC's Collaborative Computational Project for the Computer Simulation of

	Condensed Phases (CCP5) and the Molecular Simulation Group (MSG) at Daresbury Laboratory.
	`4.02` `4.03`
`dlpoly-classic`	See dlpoly. Classic is the now-retired replicated data version 2 of DLPOLY.
	`1.8(default)`
`gamess`	The General Atomic and Molecular Electronic Structure System (GAMESS) is a program for *ab initio* molecular quantum chemistry. GAMESS can compute SCF wavefunctions ranging from RHF, ROHF, UHF, GVB, & MCSCF. Correlation corrections include CI, 2nd order perturbation Theory, and Coupled-Cluster approaches, as well as the DFT approximation. Excited states can be computed by CI, EOM, or TD-DFT procedures. Nuclear gradients are available, for automatic geometry optimization, transition state searches, or reaction path following. Computation of the energy hessian permits prediction of vibrational frequencies, with IR or Raman intensities. Solvent effects may be modelled by the discrete Effective Fragment potentials, or continuum models such as the Polarizable Continuum Model.
	`20110811`
`gamess-uk`	GAMESS-UK is the general purpose *ab initio* molecular electronic structure program for performing SCF-, DFT- and MCSCF-gradient calculations, together with a variety of techniques for post Hartree Fock calculations.
	`8.0`
`Gromacs`	GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles. It is primarily designed for biochemical molecules like proteins and lipids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the non-bonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers. GROMACS supports all the usual algorithms expected from a modern MD implementation, and there are also quite a few features that make it stand out from the competition (not least performance).
	`4.5.5-double` `4.5.5-single(default)`
`lammps`	LAMMPS is a classical molecular dynamics code, and an acronym for Large-scale Atomic/Molecular Massively Parallel Simulator. LAMMPS has potentials for soft materials (biomolecules, polymers) and solid-state materials (metals, semiconductors) and coarse-grained or mesoscopic

	systems. It can be used to model atoms or, more generically, as a parallel particle simulator at the atomic, meso, or continuum scale. LAMMPS runs on single processors or in parallel using message-passing techniques and a spatial-decomposition of the simulation domain. The code is designed to be easy to modify or extend with new functionality.
	27Oct11
nwchem	NWChem software can handle, biomolecules, nanostructures, and solid-state, from quantum to classical, and all combinations, gaussian basis functions or plane-waves, scaling from one to thousands of processors, properties and relativity.
	6.0 6.0-serial
vasp	The Vienna Ab initio Simulation Package (VASP) is a computer program for atomic scale materials modelling, e.g. electronic structure calculations and quantum-mechanical molecular dynamics, from first principles. VASP computes an approximate solution to the many-body Schrödinger equation, either within density functional theory (DFT), solving the Kohn-Sham equations, or within the Hartree-Fock (HF) approximation, solving the Roothaan equations. Hybrid functionals that mix the Hartree-Fock approach with density functional theory are implemented as well. Furthermore, Green's functions methods (GW quasiparticles, and ACFDT-RPA) and many-body perturbation theory (2nd-order Møller-Plesset) are available in VASP.
	5.2(default) 5.2-debug 5.2-platform 5.2.12

13.7 Creative

Code	Description
Blender	Blender is the free open source 3D content creation suite.
	2.47(default)
MentalRay	mental ray® is a high performance, photorealistic rendering software that produces images of unsurpassed realism from computer-aided design and digital content creation data, by relying on patented and proprietary ray tracing algorithms. mental ray combines full programmability for the creation of any imaginable visual phenomenon, with setup-free interactive photorealistic rendering (iray®) in one single solution.
MentalRay-3.10	3.10.1(default)
MentalRay-3.6-3DS	3.6.51

Code	Description
DELFT3D	Delft3D is a flexible integrated modelling suite, which simulates two-dimensional (in either the horizontal or a vertical plane) and three-dimensional flow, sediment transport and morphology, waves, water quality and ecology and is capable of handling the interactions between these processes. The suite is designed for use by domain experts and non-experts alike, which may range from consultants and engineers or contractors, to regulators and government officials, all of whom are active in one or more of the stages of the design, implementation and management cycle.
	`4.00 4.00.01`
Gerris	`Gerris` is for the solution of the partial differential equations describing fluid flow.
	`121021(default)`
ncl-ncarg	NCL is an interpreted language designed for scientific data analysis and visualization.
	`6.1.0-beta(default)`
pism	Parallel Ice Sheet Model.
	`0.4(default)`
ROMS	ROMS is a free-surface, terrain-following, primitive equations ocean model widely used by the scientific community for a diverse range of applications. ROMS includes accurate and efficient physical and numerical algorithms and several coupled models for biogeochemical, bio-optical, sediment, and sea ice applications. It also includes several vertical mixing schemes, multiple levels of nesting and composed grids.
	`20110311(default)`
SWAN	SWAN is a third-generation wave model that computes random, short-crested wind-generated waves in coastal regions and inland waters.
	`40.85(default)`
TELEMAC	TELEMAC is used to simulate free-surface flows in two dimensions of horizontal space. At each point of the mesh, the program calculates the depth of water and the two velocity components.
	`v6p1(default) v6p2r1`
wps	Part of `wrf` (see below).

	3.4(default) 3.4_platform-mpi
wrf	The Weather Research & Forecasting Model.
	3.4(default) 3.4_platform-mpi
wrfda	Weather Research and Forecasting (WRF) model data assimilation system
	3.4.1(default)

13.9 Genomics (Life Sciences)

Code	Description
ABySS	Assembly By Short Sequences - a de novo, parallel, paired-end sequence assembler.
	1.2.7(default)
AmberTools	Assisted Model Building with Energy Refinement. "Amber" refers to two software stacks: a set of molecular mechanical force fields for the simulation of biomolecules (which are in the public domain, and are used in a variety of simulation programs); and a package of molecular simulation programs which includes source code and demos.
	12(default)
AmpliconNoise	AmpliconNoise is a collection of programs for the removal of noise from 454 sequenced PCR amplicons. It involves two steps the removal of noise from the sequencing itself and the removal of PCR point errors. This project also includes the Perseus algorithm for chimera removal.
	1.25(default)
BEAST	BEAST (Bayesian Evolutionary Analysis Sampling Trees) is a program for evolutionary inference of molecular sequences designed by Andrew Rambaut and Alexei Drummond (Drummond et al. 2002; 2005; 2006). It is orientated toward rooted, time-measured phylogenies inferred using molecular clock models. It can be used to reconstruct phylogenies and is also a framework for testing volutionary hypotheses without conditioning on a single tree topology. BEAST uses Bayesian MCMC analysis to average over tree space, so that each tree is weighted proportional to its posterior probability.
	1.7.1(default)

BioPerl	The Bioperl Project is an international association of developers of open source Perl tools for bioinformatics, genomics and life science.
	1.6.1(default)
BLAST	The Basic Local Alignment Search Tool (BLAST) finds regions of local similarity between sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches. BLAST can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families.
	2.2.25(default)
BLAST+	BLAST+ is a new suite of BLAST tools that utilizes the NCBI C++ Toolkit. The BLAST+ applications have a number of performance and feature improvements over the legacy BLAST applications.
	2.2.25(default)
bowtie	Bowtie is an ultrafast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small: typically about 2.2 GB for the human genome (2.9 GB for paired-end).
	0.12.7(default)
bowtie2	Bowtie 2 is an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. It is particularly good at aligning reads of about 50 up to 100s or 1,000s of characters to relatively long (e.g. mammalian) genomes, and contains a number of enhancements compared to its predecessor, bowtie.
	2.0.2(default)
BWA	Burrows-Wheeler Aligner (BWA) is an efficient program that aligns relatively short nucleotide sequences against a long reference sequence such as the human genome. It implements two algorithms, bwa-short and BWA-SW. The former works for query sequences shorter than 200bp and the latter for longer sequences up to around 100kbp. Both algorithms do gapped alignment. They are usually more accurate and faster on queries with low error rates.
	0.5.9(default)
CABOG	Celera Assembler is scientific software for biological research. Celera Assembler is a de novo whole-genome shotgun (WGS) DNA sequence assembler. The revised pipeline called CABOG (Celera Assembler with the Best Overlap Graph) is robust to homopolymer run length

	uncertainty, high read coverage, and heterogeneous read lengths.
	6.1(default)
clustalw	Multiple sequence alignment program that uses seeded guide trees and HMM profile-profile techniques to generate alignments.
	2.1(default)
clustalw-mpi	Parallelized version of ClustalW.
	0.13(default)
Curves	Curves+ is a complete rewrite of the Curves approach for analysing the structure of nucleic acids. It respects the international conventions for nucleic acid analysis, runs much faster and provides new data, notably on groove geometries. It also avoids confusion created in earlier studies between so-called "local" and "global" parameters.
	1.3
dialign	DIALIGN is a software program for multiple sequence alignment developed by Burkhard Morgenstern et al. While standard alignment methods rely on comparing single residues and imposing gap penalties, DIALIGN constructs pairwise and multiple alignments by comparing entire segments of the sequences. No gap penalty is sed. This approach can be used for both global and local alignment, but it is particularly successful in situations where sequences share only local homologies.
	2.2.1(default)
dialign-tx	DIALIGN-TX is an update to DIALIGN. See dialign.
	1.0.2(default)
eigenstrat	EIGENSTRAT (Price et al. 2006) detects and corrects for population stratification in genome-wide association studies. The method, based on principal components analysis, explicitly models ancestry differences between cases and controls along continuous axes of variation. The resulting correction is specific to a candidate marker's variation in frequency across ancestral populations, minimizing spurious associations while maximizing power to detect true associations. The approach is powerful as well as fast, and can easily be applied to disease studies with hundreds of thousands of markers.
	3.0(default) 4.2

fasttree	FastTree, a tool for inferring neighbour joining trees from large alignments. FastTree is capable of computing trees for tens to hundreds of thousands of protein or nucleotide sequences.
	2.1.3(default)
GATKSuite	The GATK is a structured software library that makes writing efficient analysis tools using next-generation sequencing data very easy, and second it's a suite of tools for working with human medical resequencing projects such as 1000 Genomes and The Cancer Gnome Atlas. These tools include things like a depth of coverage analyzers, quality score recalibrator, a SNP/indel caller and a local realigner.
	1.1.23(default)
GeneMarkS	A family of gene prediction programs developed at Georgia Institute of Technology, Atlanta, Georgia, USA.
	4.6b(default)
HMMER	HMMER is a software implementation of profile HMMs for biological sequences.
	3.0(default)
impute	IMPUTE is a program for estimating ("imputing") unobserved genotypes in SNP association studies. The program is designed to work seamlessly with the output of the genotype calling program CHIAMO and the population genetic simulator HAPGEN, and it produces output that can be analysed using the program SNPTEST.
	2.1.2(default)
JAGS	JAGS is "Just Another Gibbs Sampler". It is a program for the statistical analysis of Bayesian hierarchical models by Markov Chain Monte Carlo.
	3.2.0(default)
Kalign	Kalign - Multiple Sequence Alignment. A fast and accurate multiple sequence alignment algorithm.
	2.03(default)
LAGAN	The Lagan Tookit is a set of alignment programs for comparative genomics. The three main components are a pairwise aligner (LAGAN), a multiple aligner (M-LAGAN), and a glocalaligner (Shuffle-LAGAN). All three are based on the CHAOS local alignment tool and combine

	speed (regions up to several megabases can be aligned in minutes) with high accuracy.
	2.0(default)
lastz	LASTZ sequence alignment program. LASTZ is a drop-in replacement for BLASTZ, and is backward compatible with BLASTZ's command-line syntax. That is, it supports all of BLASTZ's options but also has additional ones, and may produce slightly different alignment results.
	1.02.00(default)
mach	Used to infer genotypes at untyped markers in genome-wide association scans.
	1.0.17(default)
mach2dat	Statistical Genetics Package.
	1.0.19(default)
mafft	Part of BioPerl.
	6.864(default)
maq	Maq stands for Mapping and Assembly with Quality It builds assembly by mapping short reads to reference sequences.
	0.7.1(default)
molquest	MolQuest is a comprehensive, easy-to-use desktop application for sequence analysis and molecular biology data management.
	2.3.3(default)
mpiBLAST	MPI version of blast.
	1.6.0(default)
mrbayes	MrBayes (Ronquist and Huelsenbeck 2003) is a program for doing Bayesian phylogenetic analysis.
	3.2.0(default)
muscle	MUSCLE (multiple sequence comparison by log-expectation) is public domain, multiple sequence alignment software for protein and nucleotide sequences. MUSCLE is often used as a replacement for Clustal, since it typically (but not always) gives better sequence alignments; in addition, MUSCLE is significantly faster than Clustal, especially for larger alignments (Edgar 2004).

	3.3(default) 3.8.31
openbugs	Openbugs - Bayesian inference Using Gibbs Sampling.
	3.2.1(default)
pauprat	PAUPRat: A tool to implement Parsimony Ratchet searches using PAUP.
	03Feb2011(default)
plink	PLINK is a free, open-source whole genome association analysis toolset, designed to perform a range of basic, large-scale analyses in a computationally efficient manner. The focus of PLINK is purely on analysis of genotype/phenotype data, so there is no support for steps prior to this (e.g. study design and planning, generating genotype or CNV calls from raw data). Through integration with gPLINK and Haploview, there is some support for the subsequent visualization, annotation and storage of results.
	1.07(default)
prank	Software for structure alignments with INFERNAL and genomic alignments.
	100802(default)
pynast	PyNAST: a tool for aligning sequences to a template alignment protocol.
	1.1(default)
qiime	QIIME (canonically pronounced 'Chime') is a pipeline for performing microbial community analysis that integrates many third party tools which have become standard in the field.
	1.3.0(default)
R_adegenet	adegenet is an R package dedicated to the exploratory analysis of genetic data. It implements a set of tools ranging from multivariate methods to spatial genetics and genome-wise SNP data analysis.
	1.3-3(default)
R_ade4	ade4: Analysis of Ecological Data: Exploratory and Euclidean methods in Environmental sciences. Multivariate data analysis and graphical display.
	1.4-17(default)

R_ape	R package - ape provides functions for reading, writing, plotting, and manipulating phylogenetic trees
	2.8(default)
R_Biostrings	Memory efficient string containers, string matching algorithms, and other utilities, for fast manipulation of large biological sequences or sets of sequences.
	2.22.0
R_gee	gee: Generalized Estimation Equation solver. Generalized Estimation Equation solver.
	4.13-17(default)
R_Geneland	Stochastic simulation and MCMC inference of structure from genetic data.
	4.0.3(default)
R_MASS	R package – mass. Software and datasets to support 'Modern Applied Statistics with S', fourth edition, by W. N. Venables and B. D. Ripley. Springer, 2002, ISBN 0-387-95457-0.
	7.3-16(default)
R_pegas	Genetic variation of the mitochondrial DNA genome.
	0.4(default)
R_seqinr	Exploratory data analysis and data visualization for biological sequence (DNA and protein) data. Include also utilities for sequence data management under the ACNUC system.
	3.0-6
R_spider	R spider implements the Global Network statistical framework to analyse gene list using as reference knowledge a global gene network constructed by combining signalling and metabolic pathways from Reactome and KEGG databases. Reactome is an expert authored, peer-reviewed knowledgebase of human reactions and pathways. Reactome database model specifies protein-protein interaction pairs. The meaning of "interaction" is broad: 2 protein sequences occur in the same complex or they occur in the same or neighbouring reaction(s). Both, Reactome signalling network and KEGG metabolic network were united into the integral network. For the human genome, the resulting integral network covers about 4000 genes involved in approximately 50,000 unique pairwise gene interactions.
	1.1-1(default)

raxml	A Tool for computing TeraByte Phylogenies.
	7.2.8(default)
rdp_classifier	The Ribosomal Database Project (RDP) provides ribosome related data and services to the scientific community, including online data analysis and aligned and annotated Bacterial and Archaeal small-subunit 16S rRNA sequences.
	2.2(default)
rmblast	RMBlast is a RepeatMasker compatible version of the standard NCBI BLAST suite. The primary difference between this distribution and the NCBI distribution is the addition of a new program "rmblastn" for use with RepeatMasker and RepeatModeler. RMBlast supports RepeatMasker searches by adding a few necessary features to the stock NCBI blastn program.
	1.2(default)
SAMtools	SAM (Sequence Alignment/Map) format is a generic format for storing large nucleotide sequence alignments.
	3.5(default)
SHRiMP	SHRiMP is a software package for aligning genomic reads against a target genome. It was primarily developed with the multitudinous short reads of next generation sequencing machines in mind, as well as Applied Biosystem's colourspace genomic representation.
	2.2.0(default)
SOAP2	SOAP has been in evolution from a single alignment tool to a tool package that provides full solution to next generation sequencing data analysis.
	2.21(default)
Spines	Spines is a collection of software tools, developed and used by the Vertebrate Genome Biology Group at the Broad Institute. It provides basic data structures for efficient data manipulation (mostly genomic sequences, alignments, variation etc.), as well as specialized tool sets for various analyses. It also features three sequence alignment packages: Satsuma, a highly parallelized program for high-sensitivity, genome-wide synteny; Papaya, an all-purpose alignment tool for less diverged sequences; and SLAP, a context-sensitive local aligner for diverged sequences with large gaps.
	1.15(default)

T-Coffee	T-Coffee is a multiple sequence alignment package. You can use T-Coffee to align sequences or to combine the output of your favourite alignment methods (Clustal, Mafft, Probcons, Muscle...) into one unique alignment (M-Coffee). T-Coffee can align Protein, DNA and RNA sequences. It is also able to combine sequence information with protein structural information (3D-Coffee/Expresso), profile information (PSI-Coffee) or RNA secondary structures (R-Coffee).
	9.02.r1228(default)
transalign	transAlign: using amino acids to facilitate the multiple alignment of protein-coding DNA sequences.
	1.2(default)
trf	Used for Detecting short tandem repeats from genome data.
	4.0.4(default)
uclust	uclust is a clustering, alignment and search algorithm capable of handling millions of sequences.
	1.2.22(default)
velvet	Velvet: algorithms for de novo short read assembly using de Bruijn graphs.
	1.1.06(default)
VMD	VMD (Visual Molecular Dynamics) is a molecular visualization program with 3-D graphics and built-in scripting for displaying, animating, and analysing large bimolecular systems.
	1.9.1

13.10 Benchmarks

These applications are mostly used by the technical team and are unlikely to offer a great deal for most users.

Benchmark	Description
imb	Intel® MPI Benchmarks.
	3.2(default)
iozone	IOzone is a filesystem benchmark tool. The benchmark generates and measures a variety of file operations. IOzone has been ported to many

	machines and runs under many operating systems, and is useful for performing a broad filesystem analysis of a vendor's computer platform. The benchmark tests file I/O performance for the following operations: Read, write, re-read, re-write, read backwards, read strided, fread, fwrite, random read, pread ,mmap, aio_read, aio_write.
	`modulefile`
`mpi_nxnlatbw`	Used for measuring port to port latency.
	`0.0`
`stream`	The `STREAM` benchmark is a simple synthetic benchmark program that measures sustainable memory bandwidth (in MB/s) and the corresponding computation rate for simple vector kernels.
	`5.6(default)`